B. Tech. Degree

in

COMPUTER SCIENCE AND ENGINEERING

SYLLABUS FOR

FLEXIBLE CURRICULUM

(For students admitted in 2022-23)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

NATIONAL INSTITUTE OF TECHNOLOGY

TIRUCHIRAPPALLI – 620 015

TAMIL NADU, INDIA.

CSE Department Vision and Mission

Department Vision

- To evolve as an internationally recognised centre of excellence for teaching and research in computer science and engineering with societal and industry relevance.

Department Mission

- To offer multidisciplinary and interdisciplinary undergraduate, postgraduate and research programmes with focus on societal research and industrial needs.
- To provide a conducive environment for learning, leading to be efficient employee's and successful entrepreneurs.
- To establish strong and solution-oriented industry academia binding for sustainable growth.
- To inculcate ethical values for holistic nation building.

Programme Educational Objectives (PEO)

- Graduates are prepared to be employed in IT industries and be engaged in continuous learning, understanding, and applying new ideas.
- Graduates are prepared to take up Masters and Research programmes.
- Graduates are prepared to be responsible computing professionals in their own area of interest solving societal problems and be successful entrepreneurs.

Programme Outcome (PO)

1. Ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modelling and design of computer based systems.
2. Ability to analyse existing literature in computer science and suggest modification to algorithms and create new algorithms for solving engineering problems.
3. Ability to apply the computing knowledge and propose solutions in domain such as health care, banking, finance, agriculture and other allied professions.
4. Ability to design software systems based on interpreting available data in all societal and engineering domains and analyse their performance.
5. Ability to analyse the problem, subdivide into smaller tasks with well-defined interface for interaction among components, and complete within the specified time frame and domain constraints.
6. Ability to propose original ideas and solutions, culminating into a modern, easy to use tool, by a larger section of the society with longevity.
7. Ability to design, implement, and evaluate secure hardware and integrate with software assuring quality and efficiency.
8. Ability to apply ethical principles and commit to professional ethics and responsibilities and norms of the computer science and engineering practices.
9. Ability to modularize the problem statement, allocate to individuals and function effectively as an individual and a team player.
10. Ability to communicate effectively the engineering solution to customers/users or peers by means of sufficient and necessary documentation.
11. Ability to demonstrate knowledge and understanding of computer science and engineering principles and apply these to individual's and team work.
12. Ability to understand contemporary issues and to get engaged in lifelong learning by independently and continually expanding knowledge and abilities.

CURRICULUM

The total minimum credits required for completing the B. Tech. programme in Computer Science and Engineering is **159.**

MINIMUM CREDIT REQUIREMENT FOR THE VARIOUS COURSE CATEGORIES:

Table 1 Course Structure

Course Category	Courses	No. of Credits	Weightage (%)
GIR (General Institute Requirement Courses)	22	50	31.25
PC (Programme Core)*	15	51	32.07
Programme Electives (PE) / Open Electives (OE)$	14	42	26.41
Essential Laboratory Requirements (ELR)	8	16	10.1
Total		**159**	**100**
Minor (Optional)	Courses for 15 credits	15 Additional credits	-
Honours (Optional)	Courses for 15 credits	15 Additional credits	-

*Minimum of 4 programme core courses shall be 4 credits each
$Out of 14 elective courses (PE/OE), the students should study at least eight programme elective courses (PE)

Table 2 GIR Courses

Sl. No.	Name of the Course	Number of Courses	Max. Credits
1.	Mathematics	3	10
2.	Physics	1	3
	Physics Lab	1	2
3.	Chemistry	1	3
	Chemistry Lab	1	2
4.	Industrial Economics and Foreign Trade	1	3
5.	English for Communication	2	4
6.	Energy and Environmental Engineering	1	2
7.	Professional Ethics	1	3
8.	Engineering Graphics	1	3
9.	Engineering Practice	1	2
10.	Basic Engineering	2	4
11.	Introduction to Computer Programming	1	3
12.	Branch Specific Course (Introduction to the Branch of Study)	1	2
13.	Summer Internship	1	2
14.	Project Work	-	-

15.	Comprehensive Viva	1	1
16.	Industrial Lecture	1	1
17.	NSS/ NCC/ NSO	1	Compulsory participation
	Total	**22**	**50**

(I) GENERAL INSTITUTE REQUIREMENTS (COURSE AND COURSE DETAILS)

1. MATHEMATICS

Sl. No.	Course Code	Course Title	Credits
1.	MAIR12	Linear Algebra and Calculus	3
2.	MAIR22	Complex Analysis and Differential Equations	3
3.	MAIR31	Mathematics III - Probability and Operations Research	4
		Total	**10**

2. PHYSICS

Sl. No.	Course Code	Course Title	Credits
1.	PHIR11	Physics	3
2.	PHIR12	Physics Lab	2
		Total	**5**

3. CHEMISTRY

Sl. No.	Course Code	Course Title	Credits
1.	CHIR11	Chemistry	3
2.	CHIR12	Chemistry Lab	2
		Total	**5**

4. HUMANITIES

Sl. No.	Course Code	Course Title	Credits
1.	HSIR13	Industrial Economics and Foreign Trade	3
		Total	**3**

5. COMMUNICATION

Sl. No.	Course Code	Course Title	Credits
1.	HSIR11 & HSIR12	English for Communication (Theory & Lab)	4
		Total	**4**

6. ENERGYAND ENVIRONMENTAL ENGINEERING

Sl. No.	Course Code	Course Title	Credits
1.	ENIR11	Energy and Environmental Engineering	2
		Total	**2**

7. PROFESSIONAL ETHICS

Sl. No.	Course Code	Course Title	Credits
1.	HSIR14	Professional Ethics	3
		Total	**3**

8. ENGINEERING GRAPHICS

Sl. No.	Course Code	Course Title	Credits
1.	MEIR12	Engineering Graphics	3
		Total	**3**

9. ENGINEERING PRACTICE

Sl. No.	Course Code	Course Title	Credits
1.	PRIR11	Engineering Practice	2
		Total	**2**

10. BASIC ENGINEERING

Sl. No.	Course Code	Course Title	Credits
1.	CEIR11	Basics of Civil Engineering	2
2.	MEIR11	Basics of Mechanical Engineering	2
		Total	**4**

11. INTRODUCTION TO COMPUTER PROGRAMMING

Sl. No.	Course Code	Course Title	Credits
1.	CSIR11	Introduction to Computer Programming (Theory and Lab)	3
		Total	**3**

12. BRANCH SPECIFIC COURSE

Sl. No.	Course Code	Course Title	Credits
1.	CSIR21	Branch Specific Course - Essentials of Computer Science	2
		Total	**2**

13. SUMMER INTERNSHIP[#]

Sl. No.	Course Code	Course Title	Credits
1.	CSIR71	Internship / Industrial Training / Academic Attachment	2
		Total	**2**

The student should undergo industrial training/internship for a minimum period of two months during the summer vacation of 3rd year. Attachment with an academic institution within the country (IISc/ IITs/ NITs/ IIITs and CFTIs) or university abroad is also permitted instead of industrial training.

[#]To be evaluated at the beginning of VII semester by assessing the report and seminar presentations.

14. INDUSTRIAL LECTURE

Sl. No.	Course Code	Course Title	Credits
1.	CSIR61	Industrial Lecture	1
		Total	**1**

A course based on industrial lectures shall be offered for 1 credit. A minimum of five lectures of two hours duration by industry experts will be arranged by the Department. The evaluation methodology, will in general, be based on quizzes at the end of each lecture.

15. COMPREHENSIVE VIVA

Sl. No.	Course Code	Course Title	Credits
1.	CSIR81	Comprehensive Viva-Voce	1
		Total	**1**

16. PROJECT WORK (OPTIONAL COURSE)

Sl. No.	Course Code	Course Title	Credits
1.	CSIR82	Project Work (Optional)	6
		Total	**6**

17. NSS / NCC / NSO

Sl. No.	Course Code	Course Title	Credits
1.	SWIR11	NSS / NCC / NSO	0
		Total	**0**

(II) PROGRAMME CORE COURSES

Sl. No.	Course Code	Course Title	Pre-requisites	Credits
1.	CSPC11	Programme Core I / Discrete Structures	-	4
2.	CSPC31	Programme Core II / Principles of Programming Languages	-	4
3.	CSPC32	Programme Core III / Data Structures	-	3
4.	CSPC33	Programme Core IV / Digital Systems Design	-	3
5.	CSPC34	Programme Core V / Computer Organization	-	3
6.	CSPC41	Programme Core VI / Formal Languages and Automata Theory	CSPC11	4
7.	CSPC42	Programme Core VII / Design and Analysis of Algorithms	CSPC32	3
8.	CSPC43	Programme Core VIII / Operating Systems	-	3
9.	CSPC51	Programme Core IX / Computer Architecture	CSPC34	4
10.	CSPC52	Programme Core X / Database Management Systems	-	3

11.	CSPC53	Programme Core XI / Computer Networks	-	3
12.	CSPC54	Programme Core XII / Introduction to Artificial Intelligence and Machine Learning	CSPC11	4
13.	CSPC61	Programme Core XIII / Embedded Systems Architectures	CSPC51	3
14.	CSPC62	Programme Core XIV / Compiler Design	CSPC41	4
15.	CSPC63	Programme Core XV / Principles of Cryptography	-	3
			Total	**51**

(III) (a) OPEN ELECTIVE COURSES

Sl. No.	Course Code	Course Title	Pre-requisites	Credits
1.	CSOE11	Big Data Analytics	-	3
2.	CSOE12	Cloud & Grid Computing	-	3
3.	CSOE13	Computer Graphics and Multimedia Systems	-	3
4.	CSOE14	Distributed Architecture	-	3
5.	CSOE15	Human Computer Interaction	-	3
6.	CSOE16	Image Processing	-	3
7.	CSOE17	Internet of Things	-	3
8.	CSOE18	Machine learning for Engineering Applications	-	3
9.	CSOE19	Security Principles	-	3
10.	CSOE20	Soft Computing	-	3
11.	CSOE21	Software Project Management	-	3
12.	CSOE22	Software Testing & Practices	-	3
13.	CSOE23	Web Technology	-	3

(b) PROGRAMME ELECTIVE COURSES (PE)

Sl. No.	Course Code	Course Title	Pre-requisites	Credits
1.	CSPE31	Linear Algebra and Applications	-	3
2.	CSPE32	Combinatorics and Graph Theory	CSPC11	3
3.	CSPE41	Software Engineering	-	3
4.	CSPE42	Design Thinking	-	3

5.	CSPE43	Advanced Data Structures and Algorithms	CSPC31	3
6.	CSPE44	Computer Graphics	-	3
7.	CSPE45	Multimedia Systems	-	3
8.	CSPE46	Computing algorithms based on Indian Knowledge Systems	-	3
9.	CSPE51	Augmented and Virtual Reality	-	3
10.	CSPE52	Digital Signal Processing	-	3
11.	CSPE53	Game Theory	-	3
12.	CSPE54	Real Time Systems	CSPC43	3
13.	CSPE55	Software Testing and Automation	-	3
14.	CSPE56	Cloud Computing	CSPC43	3
15.	CSPE57	Agile Software Development	-	3
16.	CSPE58	Data Interpretation and Analysis	-	3
17.	CSPE61	Web Technology and its Applications	CSPC42	3
18.	CSPE62	Advanced Database Management Systems	CSPC52	3
19.	CSPE63	Artificial Intelligence and its Applications	CSPC54	3
20.	CSPE64	Data Analytics	CSPC52	3
21.	CSPE65	Machine Learning Techniques and Practices	CSPC54	3
22.	CSPE66	Mobile Computing	CSPC53	3
23.	CSPE67	Internetworking Protocols	CSPC53	3
24.	CSPE68	Design and Analysis of Parallel Algorithms	CSPC42	3
25.	CSPE69	Brain Computer Interface and its Applications	-	3
26.	CSPE71	Advanced Cryptography	CSPC63	3
27.	CSPE72	Deep Learning Techniques	CSPC54	3
28.	CSPE73	Natural Language Processing	CSPC62	3
29.	CSPE74	Image Processing and Applications	-	3
30.	CSPE75	Network Security	CSPC63	3
31.	CSPE76	Wireless Network Systems	CSPC53	3
32.	CSPE77	Parallel Architectures and Programming	CSPC51	3
33.	CSPE78	Information Security	-	3
34.	CSPE79	Human Computer Interaction	-	3

35.	CSPE70	Undergraduate Research 1		3
36.	CSPE81	Data Science	-	3
37.	CSPE82	GPU Computing	CSPC51	3
38.	CSPE83	Internet of Things - Principles and Practices	-	3
39.	CSPE84	Social Network Analysis	-	3
40.	CSPE85	Speech Processing Techniques	-	3
41.	CSPE86	Deep Learning Paradigms for Computer Vision	-	3
42.	CSPE80	Undergraduate Research 2		3

(IV) ESSENTIAL PROGRAMME LABORATORY REQUIREMENTS (ELR)

Sl. No.	Course Code	Course Title	Pre-/ Co-requisites	Credits
1.	CSLR31	Laboratory I / Data structures Laboratory	CSPC31	2
2.	CSLR32	Laboratory II / Digital Laboratory	CSPC32	2
3.	CSLR41	Laboratory III / Algorithms Laboratory	CSPC42	2
4.	CSLR42	Laboratory IV / Operating Systems Lab	CSPC43	2
5.	CSLR51	Laboratory V / Database Management Systems Laboratory	CSPC52	2
6.	CSLR52	Laboratory VI / Networks Laboratory	CSPC53	2
7.	CSLR61	Laboratory VII / Embedded Systems Laboratory	CSPC61	2
8.	CSLR62	Laboratory VIII / App Development Laboratory	CSPC31	2
			Total	**16**

(V) MINOR COURSES (MI)

Sl. No.	Course Code	Course Title	Pre-requisites	Credits
1.	CSMI11	Data Structures and Algorithms	-	3
2.	CSMI12	Computer Organization	-	3
3.	CSMI13	Operating Systems	-	3
4.	CSMI14	Database Management Systems	-	3
5.	CSMI15	Software Engineering	-	3
6.	CSMI16	Computer Networks	-	3
7.	CSMI17	Artificial Intelligence	-	3

(VI) ADVANCED LEVEL COURSES FOR B. TECH. (HONOURS)

Sl. No.	Course Code	Course Title	Pre-requisites	Credits
1.	CSHO11	Software Defined Networking	-	3
2.	CSHO12	Distributed Systems	CSPC41	4
3.	CSHO13	Multi-Core Programming	CSPC51	4
4.	CSHO14	Pervasive and Ubiquitous Computing	CSPC43	3
5.	CSHO15	Middleware Technologies	-	3
6.	CSHO16	Randomized Algorithms	CSPC42	4
7.	CSHO17	Big Data Mining	-	4
8.	CSHO18	Modelling and Simulation	-	4
9.	CSHO19	Quantum Computing	-	4

Semester wise Curriculum Structure

Semester I (July Session)

Sl. No.	Course Code	Course	Credits	Category
1.	ENIR11	Energy and Environmental Engineering	2	GIR
2.	MAIR12	Linear Algebra and Calculus	3	GIR
3.	PHIR11	Physics	3	GIR
4.	PHIR12	Physics Lab	2	GIR
5.	CSIR11	Introduction to Computer Programming (Theory & Lab)	3	GIR
6.	MEIR11	Basics of Mechanical Engineering	2	GIR
7.	CEIR11	Basics of Civil Engineering	2	GIR
8.	PRIR11	Engineering Practice	2	GIR
		Total	**19**	

Semester II (January Session)

Sl. No.	Course Code	Course	Credits	Category
1.	MAIR22	Mathematics II – Complex Analysis and Differential Equations	3	GIR
2.	CHIR11	Chemistry	3	GIR
3.	CHIR12	Chemistry Lab	2	GIR
4.	HSIR11	English for Communication (Theory)	2	GIR
5.	HSIR12	English for Communication (Lab)	2	GIR
6.	CSPC11	Programme Core I / Discrete Structures	4	PC
7.	CSIR21	Branch Specific Course – Essentials of Computer Science	2	GIR
8.	MEIR12	Engineering Graphics	3	GIR
9.	SWIR11	NSS/ NCC/ NSO	0	GIR
		Total	**21**	

Semester III (July Session)

Sl. No.	Course Code	Course	Credits	Category
1.	MAIR31	Mathematics III – Probability and Operations Research	4	GIR
2.	CSPC31	Programme Core II / Principles of Programming Languages	4	PC
3.	CSPC32	Programme Core III / Data Structures	3	PC
4.	CSPC33	Programme Core IV / Digital Systems Design	3	PC
5.	CSPC34	Programme Core V / Computer Organization	3	PC
6.	CSLR31	Laboratory I / Data Structures Laboratory	2	PL
7.	CSLR32	Laboratory II / Digital Laboratory	2	PL
8.		Elective I	3	PE
		Total	**24**	

Note: Department(s) to offer Minor (MI) Course, and ONLINE Course (OC) to those willing students in addition to 24 credits.

Semester IV (January Session)

Sl. No.	Course Code	Course	Credits	Category
1.	HSIR13	Industrial Economics and Foreign Trades	3	GIR
2.	CSPC41	Programme Core VI / Formal Languages and Automata Theory	4	PC
3.	CSPC42	Programme Core VII / Design and Analysis of Algorithms	3	PC
4.	CSPC43	Programme Core VIII / Operating Systems	3	PC
5.	CSLR41	Laboratory III / Algorithms Laboratory	2	PL
6.	CSLR42	Laboratory IV / Operating Systems Lab	2	PL
7.		Elective II	3	PE
8.		Elective III	3	PE
		Total	**23**	

Note: Department(s) to offer MI/PE/OE/OC and Honours course as 2/3 credits to those willing students in addition to 23 credits.

Semester V (July Session)

Sl. No.	Course Code	Course	Credits	Category
1.	CSPC51	Programme Core IX / Computer Architecture	4	PC
2.	CSPC52	Programme Core X / Database Management Systems	3	PC
3.	CSPC53	Programme Core XI / Computer Networks	3	PC
4.	CSPC54	Programme Core XII / Introduction to Artificial Intelligence and Machine Learning	4	PC
5.	CSLR51	Laboratory V / Database Management Systems Laboratory	2	PL
6.	CSLR52	Laboratory VI / Networks Laboratory	2	PL
7.		Elective IV	3	PE
8.		Elective V	3	PE/OE
		Total	**24**	

Note: Department(s) to offer MI/PE/OE/OC and Honours course as 2/3 credits to those willing students in addition to 24 credits.

Semester VI (January Session)

Sl. No.	Course Code	Course	Credits	Category
1.	CSIR61	Industrial Lecture	1	GIR
2.	CSPC61	Programme Core XIII / Embedded Systems Architectures	3	PC
3.	CSPC62	Programme Core XIV / Compiler Design	4	PC
4.	CSPC63	Programme Core XV / Principles of Cryptography	3	PC
5.	CSLR61	Laboratory VII / Embedded Systems Laboratory	2	PL
6.	CSLR62	Laboratory VIII / App Development Laboratory	2	PL
7.	HSIR14	Professional Ethics	3	GIR
8.		Elective VI	3	PE
9.		Elective VII	3	PE/OE
		Total	**24**	

Note: Department(s) to offer MI/PE/OE/OC and Honours course as 2/3 credits to those willing students addition to 24 credits.

Semester VII (July Session)

Sl. No.	Course Code	Course	Credits	Category
1.	CSIR71	Summer Internship	2	GIR
2.		Elective VIII	3	PE
3.		Elective IX	3	PE
4.		Elective X	3	PE/OE
5.		Elective XI	3	PE/OE
		Total	**14**	

Note: Department(s) to offer MI/PE/OE/OC and Honours course as 2/3 credits to those willing students in addition to 14 credits.

Semester VIII (January Session)

Sl. No.	Course Code	Course	Credits	Category
1.	CSIR81	Comprehensive Viva-Voce	1	GIR
2.	CSIR82	Project Work[$] / Equivalent No. of Electives	6	Optional
3.		Elective XII	3	PE
4.		Elective XIII	3	PE
5.		Elective XIV	3	PE/OE
		Total	**10**	

Note: Department(s) to offer MI/PE/OE/OC and Honours course as 2/3 credits to those willing students in addition to 10 credits.

[$]Optional Course

Credit Distribution

Semester	I	II	III	IV	V	VI	VII	VIII	Total
Credit	19	21	24	23	24	24	14	10	159

Note:

1. Minimum of 4 programme core courses shall be 4 credits each.
2. Out of 14 elective courses (PE/OE), the students should study **at least eight programme elective courses (PE).**
3. MI – Minor Degree: **15 credits over and above** the minimum credit as specified by the departments. The details of MINOR will be mentioned only in the transcript not in the Degree certificate.
4. HO – Honours Degree: **15 credits over and above** the minimum credit as specified by the departments. The project work is not compulsory.

LIST OF ELECTIVES

III SEMESTER

PROGRAMME ELECTIVE 1

Course Code	Course Name	L	T	P	C
CSPE31	Linear Algebra and Applications	3	0	0	3
CSPE32	Combinatorics and Graph Theory	3	0	0	3

IV SEMESTER

PROGRAMME ELECTIVE 2 & 3

Course Code	Course Name	L	T	P	C
CSPE41	Software Engineering	3	0	0	3
CSPE42	Design Thinking	3	0	0	3
CSPE43	Advanced Data Structures and Algorithms	3	0	0	3
CSPE44	Computer Graphics	3	0	0	3
CSPE45	Multimedia Systems	3	0	0	3
CSPE46	Computing Algorithms based on Indian Knowledge Systems	3	0	0	3

V SEMESTER

PROGRAMME ELECTIVE 4 & 5

Course Code	Course Name	L	T	P	C
CSPE51	Augmented and Virtual Reality	3	0	0	3
CSPE52	Digital Signal Processing	3	0	0	3
CSPE53	Game Theory	3	0	0	3
CSPE54	Real Time Systems	3	0	0	3
CSPE55	Software Testing and Automation	3	0	0	3
CSPE56	Cloud Computing	3	0	0	3
CSPE57	Agile Software Development	3	0	0	3
CSPE58	Data Interpretation and Analysis	3	0	0	3

VI SEMESTER

PROGRAMME ELECTIVE 6 & 7

Course Code	Course Name	L	T	P	C
CSPE61	Web Technology and its Applications	3	0	0	3
CSPE62	Advanced Database Management Systems	3	0	0	3
CSPE63	Artificial Intelligence and its Applications	3	0	0	3
CSPE64	Data Analytics	3	0	0	3
CSPE65	Machine Learning Techniques and Practices	3	0	0	3
CSPE66	Mobile Computing	3	0	0	3
CSPE67	Internetworking Protocols	3	0	0	3
CSPE68	Design and Analysis of Parallel Algorithms	3	0	0	3
CSPE69	Brain Computer Interface and its Applications	3	0	0	3

VII SEMESTER

PROGRAMME ELECTIVE 8, 9, 10 & 11

Course Code	Course Name	L	T	P	C
CSPE71	Advanced Cryptography	3	0	0	3
CSPE72	Deep Learning Techniques	3	0	0	3
CSPE73	Natural Language Processing	3	0	0	3
CSPE74	Image Processing and Applications	3	0	0	3
CSPE75	Network Security	3	0	0	3
CSPE76	Wireless Network Systems	3	0	0	3
CSPE77	Parallel Architectures and Programming	3	0	0	3
CSPE78	Information Security	3	0	0	3
CSPE79	Human Computer Interaction	3	0	0	3
CSPE70	Undergraduate Research 1	0	0	6	3

VIII SEMESTER

PROGRAMME ELECTIVE 12, 13 & 14

Course Code	Course Name	L	T	P	C
CSPE81	Data Science	3	0	0	3
CSPE82	GPU Computing	3	0	0	3
CSPE83	Internet of Things - Principles and Practices	3	0	0	3
CSPE84	Social Network Analysis	3	0	0	3

CSPE85	Speech Processing Techniques	3	0	0	3
CSPE86	Deep Learning Paradigms for Computer Vision	3	0	0	3
CSPE80	Undergraduate Research 2	0	0	6	3

MINORS

Course Code	Course Name	L	T	P	C
CSMI11	Data Structures and Algorithms	3	0	0	3
CSMI12	Computer Organization	3	0	0	3
CSMI13	Operating Systems	3	0	0	3
CSMI14	Database Management Systems	3	0	0	3
CSMI15	Software Engineering	3	0	0	3
CSMI16	Computer Networks	3	0	0	3
CSMI17	Artificial Intelligence	3	0	0	3

HONOURS

Course Code	Course Name	L	T	P	C
CSHO11	Software Defined Networking	3	0	0	3
CSHO12	Distributed Systems	3	1	0	4
CSHO13	Multi-Core Programming	3	1	0	4
CSHO14	Pervasive and Ubiquitous Computing	3	0	0	3
CSHO15	Middleware Technologies	3	0	0	3
CSHO16	Randomized Algorithms	3	1	0	4
CSHO17	Big Data Mining	3	1	0	4
CSHO18	Modelling and Simulation	3	1	0	4
CSHO19	Quantum Computing	3	1	0	4

OPEN ELECTIVES

Course Code	Course Name	L	T	P	C
CSOE11	Big Data Analytics	3	0	0	3
CSOE12	Cloud & Grid Computing	3	0	0	3
CSOE13	Computer Graphics and Multimedia Systems	3	0	0	3
CSOE14	Distributed Architecture	3	0	0	3

CSOE15	Human Computer Interaction	3	0	0	3
CSOE16	Image Processing	3	0	0	3
CSOE17	Internet of Things	3	0	0	3
CSOE18	Machine Learning for Engineering Applications	3	0	0	3
CSOE19	Security Principles	3	0	0	3
CSOE20	Soft Computing	3	0	0	3
CSOE21	Software Project Management	3	0	0	3
CSOE22	Software Testing & Practices	3	0	0	3
CSOE23	Web Technology	3	0	0	3

LIST OF ONLINE COURSES

1. Fundamentals of Image and Video Processing
2. Deep Learning in Computer Vision
3. Deep Learning in Python
4. Image Processing and Analysis for Life Scientist
5. Interactive Computer Graphics
6. Practical Machine Learning
7. Software Security
8. Introduction to Cyber Security
9. Machine Learning: Regression
10. Cyber Security Fundamentals
11. Neural Networks and Deep Learning
12. Text Retrieval and Search Engines
13. Discrete Optimization
14. The Arduino Platform and C Programming
15. Introduction to the Internet of Things and Embedded Systems
16. Computational Neuroscience
17. Introduction to Software Product Management
18. Software Processes and Agile Practices
19. The Raspberry Pi Platform and Python Programming for the Raspberry Pi
20. Introduction to Recommender Systems: Non-Personalized and Content-Based
21. System Validation: Model Process Behaviour
22. Advanced Modelling for Discrete Optimization
23. Improving Deep Neural Networks: Hyper parameter Tuning, Regularization and Optimization
24. Big Data Applications: Machine Learning at Scale
25. Introduction to Cyber security for Business
26. Block Chain Architecture Design and Use Cases

SYLLABUS

II SEMESTER

Course Code	:	**CSPC11**
Course Title	:	**Discrete Structures**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PC**

Course Objectives

- To get familiar and understand the fundamental notions in discrete mathematics
- To describe binary relations between two sets; determine if a binary relation is reflexive, symmetric, or transitive or is an equivalence relation; combine relations using set operations and composition
- To understand and demonstrate the basic concept of an algorithm and its application in combinatorial mathematics
- To identify the base step and the recursive or inductive step in applied problems and give a recursive and a non-recursive definition for an iterative algorithm
- To identify the basic properties of graphs and trees and model simple applications

Course Contents

UNIT I Set Theory and Logic

Sets - Functions - Relations - Equivalence Relation - Poset - Functions Logic: Propositional Logic - Truth Tables - Tautologies - Resolution Proof System - Predicate Logic.

UNIT II Induction and Combinatorics

Peano's Axioms - Mathematical Induction - Pigeon Hole Principle - Principle of Inclusion and Exclusion - Review of Permutations and Combinations - Distribution Problems - Derangements - Bijection Principle.

UNIT III Algebraic Structures

Semi-Groups - Monoids - Groups - Subgroups and their properties - Cyclic groups - Cosets - Permutation Groups - Lagrange's Theorem - Cayley's Theorem - Normal Subgroups - Homomorphism of Groups - Quotient Groups - Introduction to Rings and Fields.

UNIT IV Linear Algebra and Recurrence relations

Linear Algebra: Vector space - Basis - Dimension - Orthogonally - Recurrence Relations: Homogenous and Inhomogenous - Recurrences and their solutions - Solving Recurrences using Generating functions.

UNIT V Graph Theory

Definitions and basic results - Representation of a graph by a matrix and Adjacency list - Trees - Cycles - Properties - Paths and Connectedness - Subgraphs - Graph Isomorphism - Operations on Graphs - Vertex and Edge cuts - Vertex and Edge connectivity.

Course Outcomes

Upon completion of this course, the students will be able to:

- Distinguish between the notion of discrete and continuous mathematical structures
- Prove basic set equalities
- Construct and interpret finite state diagrams and DFSA
- Apply induction and other proof techniques towards problem solving
- Solve problems in Computer Science using graphs and trees

Text Books

1. C. L. Liu, D. P. Mohapatra, "Elements of Discrete Mathematics: A Computer oriented Approach", McGraw Hill, Third Edition, 2012.
2. Kenneth H. Rosen, "Discrete Mathematics and its Applications", Seventh Edition, McGraw Hill, 2012 *(Indian Adaptation by Kamala Krithivasan, IIT Madras)*.
3. Narsing Deo, "Graph Theory with Applications to Engineering and Computer Science", Prentice Hall India Pvt. Ltd., 1979.

Mapping of Course Outcomes with Programme Outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	0	2	2	0	0	0	0	0	0	2
CO2	3	3	0	2	2	0	0	0	0	0	0	0
CO3	3	3	2	2	0	0	0	0	0	0	0	3
CO4	3	3	0	2	0	0	0	0	0	0	0	2
CO5	3	3	2	2	0	0	0	0	0	0	0	3

Course Code	:	**CSIR21**
Course Title	:	**Essentials of Computer Science**
Number of Credits	:	**2–0–0–2**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**GIR**

Course Objectives

- To make the student understand the basic building blocks of a computing system
- To make the student understand the flow of Concept- Program-Input-Processing-Output
- To introduce low level language, translators, operating system

Course Contents

UNIT I Concept - Program - Input - Processing - Output
Demo of simple high level language program to low level machine level language program - tracing their execution from high level to circuit level/ gate level - Overview of the Hardware Description Language (HDL) - Designing a set of elementary logic gates from primitive NAND gates.
Design of binary adders - culminating in the construction of a simple ALU (Arithmetic–Logic Unit) using logic gates - Design of memory hierarchy from elementary flip–flop gates to registers and RAM units of arbitrary sizes using logic gates.

UNIT II Introduction to Low Level Language
Introducing an instruction set in both binary and assembly (symbolic) versions - Writing some low-level assembly programs - Other details of computer architecture - Basic language translation techniques: parsing - symbol table - macro - assembly

UNIT III Introduction to Virtual Machine
The role of virtual machines in modern software architectures like Java and .NET - Introduction of a typical VM language - focusing on stack–based arithmetic - logical - and memory access operations - VM abstraction and implementation - focusing on stack-based flow-of-control and subroutine call-and-return techniques.

UNIT IV Introduction to Compilers
Context-free grammars and recursive parsing algorithms - Building a syntax analyzer (tokenizer and parser) - The syntax analyzer to generate XML code reflecting the structure of the translated program - Code generation - low-level handling of arrays and objects.

UNIT V Introduction to OS
Discussion of OS/hardware and OS/software design trade–offs - and time/space efficiency considerations - Design and implementation of OS - memory management - string processing - I/O handling algorithms.

Course Outcomes

Upon completion of this course, the students will be able to:

- Trace the fundamentals of digital logic design
- Virtualize any environment
- Prepare themselves for designing a compiler
- Generate low level code for simple programs
- Design simple arithmetic and memory units

Text Books

1. Noam Nisan, Shimon Schocken, “The Elements of Computing System: Building a Modern Computer from First Principles”, MIT Press, 2005.
2. Paul Tymann, Carl Reynolds, “Principles of Computer Science”, Schaum’s Series, 2005.
3. Raman Mata-Toledo, Pauline K Cushman, “Introduction to Computer Science”, Schaum’s Series, 2000.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	3	0	0	0	0	0	2	0	1	1	0
CO2	2	0	2	0	3	3	0	1	0	0	0	1
CO3	1	0	0	2	0	2	0	0	0	0	0	2
CO4	0	3	0	2	0	2	2	0	0	0	0	2
CO5	1	0	2	0	0	0	3	3	0	0	0	0

III SEMESTER

Course Code	:	**MAIR31**
Course Title	:	**Mathematics III - Probability and Operations Research**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**GIR**

Course Objectives

- To introduce the fundamental concepts and results in probability and to learn some standard distributions
- To study one dimensional and multi-dimensional random variables
- To understand elementary queuing concepts and Markov chain and process
- To introduce linear programming problem and discuss several methods to solve it
- To study the transportation and assignment problems and to explore the project management techniques

Course Contents

UNIT I
Definitions of Probability - Conditional Probability - Baye's Rule - Random variable - Probability mass function and Density function - Binomial Distribution - Poisson Distribution - Normal Distribution - Moment Generating Function.

UNIT II
Joint Probability Density Function - Marginal and Conditional Densities - Function of Random Variable - Covariance and Conditional Expectation - Correlation Coefficient.

UNIT III
Random Process - Finite Markov Chain - Various States - Limiting Probability - Markov Process - Poisson process - M/M/l Queues with finite and infinite waiting space.

UNIT IV
Models in Operations Research - Linear Programming Problem (LPP) - Graphical Method - Simplex Method - Big-M Method - Duality - Sensitivity Analysis - Dual simplex method.

UNIT V
Transportation and Assignment Problems - Project Management and Network - Critical Path Method (CPM) - Program Evaluation and Review Technique (PERT).

Course Outcomes
Upon completion of this course, the students will be able to:

- Find the probability of an event and the moment generating function of Binomial, Poisson and Normal distributions
- Determine marginal and conditional probability distributions and correlation of two dimensional random variables
- Approximate and solve the real problems using random process and queuing theory
- Solve linear programming problems using Graphical, Simplex, Big-M and Dual Simplex methods
- Determine an optimal solution for transportation and assignment problems and determine the expected project completion time in a project network

Text Books

1. Sheldon M. Ross, S.M., "Introduction to Probability and Statistics for Engineers & Scientists", Academic press, 2009.
2. Kishor S. Trivedi, "Probability and Statistics with Reliability, Queuing and Computer Science Applications", John Wiley & Sons, 2016.

3. Arnold O. Allen, “Probability, Statistics and Queueing Theory with Computer Science Applications”, Academic Press, 2006.
4. A. Ravindran, Don T. Phillips, James J. Solberg, “Operations Research - Principles and Practice”, John Wiley, 2014.
5. Hamdy A. Taha, “Operations Research - An Introduction”, Tenth Edition, Pearson, 2017.
6. Frederick S. Hillier, Gerald J. Lieberman, “Introduction to Operations Research”, McGraw Hill, 2015.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	0	0	2	0	0	0	0	1	0	0
CO2	2	0	0	0	2	0	0	0	0	1	0	0
CO3	0	0	1	2	2	0	0	0	2	0	0	2
CO4	2	0	1	0	2	2	0	0	2	0	0	2
CO5	2	0	3	0	2	0	0	0	2	1	0	0

Course Code	:	**CSPC31**
Course Title	:	**Principles of Programming Languages**
Number of Credits	:	**3-0-2-4**
Pre-requisites (Course Code)	:	-
Course Type	:	**PC**

Course Objectives

- To understand the various ways to describe syntax and semantics of programming languages
- To understand data, data types, and basic statements of programming languages
- To understand parameter passing and function call mechanisms
- To understand object-orientation, concurrency, and event handling in programming languages
- To acquire knowledge about functional and logic programming paradigms

Course Contents

UNIT I Syntax and Semantics and Basic Statements

Evolution of programming languages - describing syntax & semantics - lexical analysis - parsing - recursive-decent bottom up parsing - primitive data types – strings - array types - associative arrays - record types - union types - pointers and references - Arithmetic expressions - relational and Boolean expressions - assignment statements - mixed-mode assignments - control structures - selection - iterations - branching - guarded statements.

Lab Component (Exercises similar to the following):

1. Simple test programs to determine type compatibility rules of a C compiler.
2. Simple test program to determine the scope of variables having the same name and different names declared within a while / for loop.
3. Program that behaves differently if name equivalence is used against structural equivalence.
4. Write a program to convert one form of comments in C to alternate comments form.

UNIT II Subprograms and Implementations

Subprograms - design issues - local referencing - parameter passing - overloaded methods - generic methods - design issues for functions - semantics of call and return - implementing simple subprograms - stack - dynamic local variables - nested subprograms - blocks - dynamic scoping.

Lab Component (Exercises similar to the following):

1. Write a program to determine the ratio of the time required to pass a large array by reference and the time required to pass the same array by value.
2. Write a program that determines whether it is legal to call a function that has been passed by passing a pointer to it to another function.
3. Devise a subprogram and calling code in which pass-by-reference and pass-by-value-result of one or more parameters produces different results.
4. Design a skeletal program and a calling sequence that results in an activation record instance in which the static and dynamic links point to different activation recorded instances in the run-time stack.

UNIT III Object-Orientation, Concurrent, and Event Driven Programming

Object-orientation design issues for OOP languages - implementation of object-oriented constructs - concurrency - semaphores - monitors - message passing - threads - statement level concurrency - exception handling - Event driven control - Event Handling

Lab Component (Exercises similar to the following):

1. Chess / checkers game using object oriented programming – C++/Smalltalk / Python / Java.
2. Design a Tic-tac-toe game that uses even driven programming concepts.
3. The bouncing ball game is one where, there are more than 2 balls bouncing around a window. When the ball reaches the edge of a window, it reverses direction. Write program using concurrent programming concepts to implement this game. Try alternate ways of changing the ball's direction.

UNIT IV Functional Programming

Introduction to lambda calculus - fundamentals of functional programming languages - Programming with Scheme - Introduction to LISP - Lists - Storage allocation for lists - Some useful functions - Error handling.

Lab Component (Exercises similar to the following):

1. Scheme functions to compute mathematical formula, roots of a quadratic equation, count of characters in a string, set computation like, Union, Intersection, complementation, etc.
2. Lisp recursive function to return 'nth' item from a list, diagonal of a matrix, sum of the diagonal of matrix, a sub-string from a string, etc.

UNIT V Logic Programming

Introduction to logic and logic programming - Computing with relations - Programming with Prolog - Data structures in Prolog - Programming techniques - Control in Prolog - Cuts - multi-paradigm languages.

Lab Component (Exercises similar to the following):

1. Prolog program to find the factorial of a number, simplification of arithmetic expression involving additive, multiplicative identity, solve Sudoku puzzle, etc.

Course Outcomes

Upon completion of this course, the students will be able to:

- Define data types, functions, syntax and semantics of all programming languages
- Use the various styles of programming languages for any given problem
- Compare and use appropriate parameter passing technique for solving problems
- Distinguish between the usage of all programming languages
- Apply the appropriate programming language to solve real-world problems

Text Books

1. Robert W. Sebesta, "Concepts of Programming Languages", Tenth Edition, Addison Wesley, 2012.
2. Michael L. Scott, "Programming Language Pragmatics", Third Edition, Morgan Kaufmann, 2009.

Reference Books

1. Allen B Tucker, and Robert E Noonan, "Programming Languages – Principles and Paradigms", Second Edition, Tata McGraw Hill, 2007.
2. R. Kent Dybvig, "The Scheme Programming Language", Fourth Edition, MIT Press, 2009.
3. Jeffrey D. Ullman, "Elements of ML Programming", Second Edition, Prentice Hall, 1998.
4. Richard A. O'Keefe, "The Craft of Prolog", MIT Press, 2009.
5. W. F. Clocksin, C. S. Mellish, "Programming in Prolog: Using the ISO Standard", Fifth Edition, Springer, 2003.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	0	0	0	0	0	0	0	0	0	0
CO2	0	0	1	1	0	2	0	0	0	0	0	0
CO3	0	2	0	0	3	0	0	0	2	0	0	0
CO4	3	0	0	0	0	0	0	0	0	1	0	1
CO5	1	0	3	2	0	3	0	0	0	0	0	0

Course Code	:	**CSPC32**
Course Title	:	**Data Structures**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PC**

Course Objectives

- To find the Time Complexity and Space Complexity for algorithm
- To understand the various techniques of sorting and searching
- To design and implement arrays, stacks, queues, and linked lists
- To understand the complex data structures such as trees and graphs
- To solve real time problems

Course Contents

UNIT I Introduction
Development of Algorithms - Notations and analysis - Storage structures for arrays - Sparse matrices - Stacks and Queues: Representations and applications.

UNIT II Linked List, Stacks, and Queues
Linked Lists - Linked stacks and queues - Operations on polynomials - Doubly linked lists - Circularly linked lists - Dynamic storage management - Garbage collection and compaction.

UNIT III Trees
Binary Trees - Binary search trees - Tree traversal - Expression manipulation - Symbol table construction - Height balanced trees - AVL trees - Red-black trees.

UNIT IV Graphs
Graphs - Representation of graphs - BFS - DFS - Topological sort - String representation and manipulations - Pattern matching.

UNIT V Sorting and Searching
Sorting Techniques - Selection - Bubble - Insertion - Merge - Heap - Quick - Radix sort - Address calculation - Linear search - Binary search - Hash table methods.

Course Outcomes

Upon completion of this course, the students will be able to:

- Develop programs to implement linear data structures such as stacks, queues, linked lists, etc.
- Apply the concept of trees and graph data structures in real world scenarios
- Appropriately to decide on the data structure for any practical problem
- Comprehend the implementation of sorting and searching algorithms
- Compare Time Complexity and Space Complexity for algorithm

Text Books

1. J. P. Tremblay, P. G. Sorenson, "An Introduction to Data Structures with Applications", Second Edition, Tata McGraw Hill, 1981.
2. M. Tenenbaum, Augestien, "Data Structures using C", Third Edition, Pearson Education, 2007.
3. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C", Second Edition, Addison-Wesley Educational Publishers, 2006.

Reference Book

1. Sartaj Sahni, “Data Structures, Algorithms and Applications in C++”, Universities Press (I) Pvt. Ltd., 2008.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	0	0	0	0	0	0	0	0	0	0
CO2	3	0	1	0	0	0	0	0	0	0	0	0
CO3	3	2	1	1	0	2	0	0	0	0	2	0
CO4	3	0	1	1	2	0	0	0	2	0	0	1
CO5	2	2	1	0	0	2	0	0	0	0	2	1

Course Code	:	**CSPC33**
Course Title	:	**Digital Systems Design**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PC**

Course Objectives

- To understand the overview on the design principles of digital computing systems
- To learn the various number systems
- To learn Boolean Algebra and Understand the various logic gates
- To be familiar with various combinational circuits
- To be familiar with designing synchronous and asynchronous sequential circuits
- To be exposed to designing using PLD

Course Contents

UNIT I Boolean Algebra
Binary codes - Weighted and non-weighted Binary arithmetic conversion algorithms - Canonical and standard boolean expressions - Truth tables - K-map reduction - Don't care conditions - Adders / Subtractors - Carry look-ahead adder - Code conversion algorithms - Design of code converters - Equivalence functions.

UNIT II Combinational Circuits
Binary/Decimal Parallel Adder/Subtractor for signed numbers - Magnitude comparator - Decoders / Encoders - Multiplexers / Demultiplexers - Boolean function implementation using multiplexers.

UNIT III Sequential Circuits
Sequential logic - Basic latch - Flip-flops (SR - D - JK - T - Master-Slave) - Triggering of flip-flops - Counters - Design procedure - Ripple counters - BCD and Binary - Synchronous counters - Registers - Shift registers - Registers with parallel load - Reduction of state and flow tables - Race-free state assignment - Hazards.

UNIT IV VLSI Design
Introduction to VLSI design - Basic gate design - Digital VLSI design - Design of general boolean circuits using CMOS gates - Verilog Concepts - Basic concepts - Modules & ports & Functions - useful modelling techniques - Timing and delays - user defined primitives - Modelling Techniques.

UNIT V Advanced Concepts
Advanced Verilog Concepts - Synthesis concepts - Inferring latches and flip-flops - Modelling techniques for efficient circuit design - Design of high-speed arithmetic circuits - Parallelism Pipelined Wallace tree tipliers - Systolic algorithms - Systolic matrix multiplication.

Course Outcomes

Upon completion of this course, the students will be able to:

- Design and implement complicated digital systems using Verilog
- Design a VLSI circuit for an application
- Comprehend the digital design logic
- Design and Analysis of a given digital circuit – combinational and sequential
- Use Boolean simplification techniques to design a combinational hardware circuit

Text Books

1. Morris Mano, and Michael D. Ciletti, “Digital Design”, Fifth Edition, PHI, 2012.
2. Samir Palnitkar, “Verilog HDL”, Second Edition, Pearson Education, 2003.

Reference Books

1. Michael D. Ciletti, “Advanced Digital Design with the Verilog HDL”, Second Edition, Pearson Education, 2010.
2. Stephen Brown, “Fundamentals of Digital Logic with Verilog”, McGraw Hill, 2007.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	1	0	2	0	2	2	2	0	0	1
CO2	3	0	1	0	2	2	2	0	0	0	0	1
CO3	2	0	1	0	0	0	2	0	0	0	0	0
CO4	3	0	0	0	0	0	2	0	0	0	0	1
CO5	2	0	0	0	2	0	2	0	0	0	0	0

Course Code	:	**CSPC34**
Course Title	:	**Computer Organization**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PC**

Course Objectives

- To understand the basic hardware and software issues of computer organization
- To understand the representation of data at machine level
- To understand how computations are performed at machine level
- To understand the memory hierarchies, cache memories and virtual memories
- To learn the different ways of communication with I/O devices

Course Contents

UNIT I
Introduction - Technologies for building Processors and Memory - Performance - The Power Wall - Operations of the Computer Hardware - Operands Signed and Unsigned numbers - Representing Instructions - Logical Operations - Instructions for Making Decisions

UNIT II
MIPS Addressing for 32-Bit Immediates and Addresses - Parallelism and Instructions: Synchronization - Translating and Starting a Program - Addition and Subtraction - Multiplication - Division - Floating Point - Parallelism and Computer Arithmetic: Subword Parallelissm - Streaming SIMD Extensions and Advanced Vector Extensions in x86.

UNIT III
Logic Design Conventions - Building a Datapath - A Simple Implementation Scheme - overview of Pipelining - Pipelined Datapath - Data Hazards: Forwarding versus Stalling - Control Hazards - Exceptions - Parallelism via Instructions - The ARM Cortex–A8 and Intel Core i7 Pipelines - Instruction–Level Parallelism and Matrix Multiply Hardware Design language.

UNIT IV
Memory Technologies - Basics of Caches - Measuring and Improving Cache Performance - dependable memory hierarchy - Virtual Machines - Virtual Memory - Using FSM to Control a Simple Cache - Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks - Advanced Material: Implementing Cache Controllers.

UNIT V
Disk Storage and Dependability - Parallelism and Memory Hierarchy: RAID levels - performance of storage systems - Introduction to multi threading clusters - message passing multiprocessors.

Course Outcomes

Upon completion of the course, the students will be able to:

- Understand the architecture and functionality of central processing unit
- Analyze the abstraction of various components of a computer
- Analyze the hardware and software issues and the interfacing
- Work out the trade-offs involved in designing a modern computer system
- Understand the various memory systems and I/O communication

Text Books

1. David A. Patterson, John L. Hennessey, “Computer Organization and Design, The Hardware/Software Interface”, Fifth Edition, Morgan Kauffman/Elsevier, 2014.
2. Smruti Ranjan Sarangi, “Computer Organization and Architecture”, McGraw Hill Education, 2015.

Reference Books

1. V. Carl Hamacher, Zvonko G. Varanesic, Safat G. Zaky, “Computer Organization”, Sixth Edition, McGraw-Hill Inc., 2012.
2. William Stallings, “Computer Organization and Architecture”, Eighth Edition, Pearson Education, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	2	0	0	0	2	0	0	0	0	0
CO2	0	0	2	0	0	0	3	0	0	0	0	0
CO3	0	0	3	0	2	0	0	0	0	0	0	0
CO4	3	3	0	0	1	0	2	0	0	2	0	3
CO5	0	0	0	0	3	0	1	0	0	0	0	0

Course Code	:	**CSLR31**
Course Title	:	**Data Structures Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**CSPC31**
Course Type	:	**PL**

Course Objectives

- To analyze the time and space complexities and efficiency of various algorithms
- To understand the practical application of linear and nonlinear data structures
- To introduce and practice advanced algorithms, programming techniques necessary for developing sophisticated computer application programs

Exercises

1. Problems in C/C++/ Java using data structures involving arrays, stacks, queues, strings, linked lists, trees, graphs.
2. Operations on stacks, queues and linked lists.
3. Applications of stack - Conversion of infix expressions to postfix and evaluation of postfix expressions.
4. Application of linked lists – Singly, Doubly.
5. Implementation of priority queue.
6. Implementation of Binary Tree, Binary Search Tree, AVL Trees.
7. Implementation of BFS, DFS – Application of stack, queues.
8. Implementation of Sorting Techniques.

Course Outcomes

Upon completion of this course, the students will be able to:

- Apply and implement the learned algorithm for problem solving
- Identify the data structure to develop program for real time applications
- Design and develop optimal algorithms using appropriate data structures
- Apply the linear/non linear data structure operations for a given problem
- Apply the hash functions with collision free operations for data storage and retrieval

Text Book

1. M. Tenenbaum, Augestien, “Data Structures using C”, Third Edition, Pearson Education, 2007.

Mapping of Course Outcomes with Programme Outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	0	2	3	0	3	0	0	0	0	3	0
CO2	2	3	3	0	3	0	0	2	0	0	3	0
CO3	3	0	1	0	3	2	0	0	3	2	0	0
CO4	2	0	2	3	0	3	0	0	0	0	2	0
CO5	3	0	1	2	0	3	0	0	0	0	3	0

Course Code	:	**CSLR32**
Course Title	:	**Digital Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**CSPC32**
Course Type	:	**PL**

Course Objectives

- To understand the overview on the design principles of digital computing systems
- To learn Boolean Algebra and Understand the various logic gates
- To develop programs in Hardware Description Language
- To design and implement synchronous sequential, asynchronous sequential circuits
- To be familiar with basic combinational and sequential components used in the typical data path designs

Exercises

1. Verification of Boolean Theorems using basic gates.
2. Design and implementation of combinational circuits for arbitrary functions and code converters.
3. Design and implement Adder and Subtractor.
4. Implement Parity generator / checker.
5. Design and implement combinational circuits: 4 –bit binary adder / subtractor.
6. Design and implement shift-registers.
7. Design and implement synchronous and asynchronous counters.
8. Coding sequential circuits using HDL.
9. Coding combinational circuits using HDL.
10. Design of a 32-bit carry look-ahead adder with logarithmic depth using Verilog.
11. Design of a Wallace tree multiplier using Verilog.
12. Design and implementation of board using Verilog.

Course Outcomes

Upon completion of this course, the students will be able to:

- Comprehend the digital design logic
- Design synchronous sequential circuits using basic flip-flops, counters, PLA, PAL
- Design and develop basic digital systems
- Debug digital circuits
- Use boolean simplification techniques to design a combinational hardware circuit

Text Books

1. Morris Mano, Michael D. Ciletti, “Digital Design”, Fifth Edition, PHI, 2012.
2. Samir Palnitkar, “Verilog HDL”, Second Edition, Pearson Education, 2003.

Reference Books

1. Michael D. Ciletti, “Advanced Digital Design with the Verilog HDL”, Second Edition, Pearson Education, 2010.
2. Stephen Brown, “Fundamentals of Digital Logic with Verilog”, McGraw Hill, 2007.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	1	0	2	2	2	0	1	0	0	0
CO2	2	0	2	0	3	3	3	0	1	0	0	0
CO3	3	3	3	1	0	2	3	0	1	0	0	0
CO4	0	1	0	0	0	1	2	0	0	0	0	0
CO5	3	0	1	0	0	1	2	0	0	0	0	0

IV SEMESTER

Course Code	:	**CSPC41**
Course Title	:	**Formal Languages and Automata Theory**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**CSPC11**
Course Type	:	**PC**

Course Objectives

- To know about Chomsky hierarchy for organizing languages
- To introduce concepts in automata theory and theory of computation
- To identify different formal language classes and their relationships
- To design grammars and recognizers for different formal languages
- To understand undecidability and decide on languages that are undecidable

Course Contents

UNIT I Finite Automata
Alphabets - Strings and Languages - Automata and Grammars - Deterministic Finite Automata (DFA) - Formal Definition - Simplified notation: State transition graph - Transition table - Language of DFA - Nondeterministic Finite Automata (NFA) - NFA with epsilon transition - Language of NFA - Equivalence of NFA and DFA - Minimization of Finite Automata - Distinguishing one string from other - Myhill-Nerode Theorem.

UNIT II Regular Expression (RE)
Definition - Operators of regular expression and their precedence - Algebraic laws for Regular expressions - Kleen's Theorem - Regular expression to FA - DFA to Regular expression - Arden Theorem - Non Regular Languages - Pumping Lemma for regular Languages. Application of Pumping Lemma - Closure properties of Regular Languages - Decision properties of Regular Languages - FA with output: Moore and Mealy machine - Equivalence of Moore and Mealy Machine - Applications and Limitation of FA.

UNIT III Context Free Grammar (CFG) and Context Free Languages
Definition - Examples - Derivation - Derivation trees - Ambiguity in Grammar - Inherent ambiguity - Ambiguous to Unambiguous CFG - Useless symbols - Simplification of CFGs - Normal forms for CFGs: CNF and GNF - Closure properties of CFLs - Decision Properties of CFLs: Emptiness - Finiteness and Membership - Pumping lemma for CFLs.

UNIT IV Push Down Automata (PDA)
Description and definition - Instantaneous Description - Language of PDA - Acceptance by Final state - Acceptance by empty stack - Deterministic PDA - Equivalence of acceptance by empty stack and final state - Conversion of CFG to PDA and PDA to CFG.

UNIT V Turing Machines (TM) and Undecidability
Basic model - definition and representation - Instantaneous Description - Language acceptance by TM - Variants of Turing Machine - TM as Computer of Integer functions - Universal TM - Church's Thesis - Recursive and recursively enumerable languages - Halting problem - Introduction to Undecidability - Undecidable problems about TMs - Post correspondence problem (PCP) - Modified PCP and undecidable nature of post correspondence problem - Introduction to recursive function theory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Design finite automata or regular expression for any tokenization task
- Construct a context free grammar for parsing any language
- Design Turing machine for any language
- Conclude the decidable / undecidable nature of any language
- Apply mathematical and formal techniques for solving real-world problems

Text Book

1. John Hopcroft, Rajeev Motwani, and Jeffrey Ullman, "Introduction to Automata Theory, Languages and Computation", Third Edition, Pearson Education, 2014.

Reference Books

1. John Hopcroft, Jeffrey Ullman, "Introduction to Automata Theory, Languages and Computation", Nineteenth Reprint, Narosa Publishing House, 2002.
2. Martin J. C., "Introduction to Languages and Theory of Computations", Fourth Edition, TMH, 2010.
3. Peter Linz, "An Introduction to Formal Language and Automata", Narosa Pub. House, 2011.
4. Papadimitriou C., Lewis C. L., "Elements of the Theory of Computation", PHI, 1997.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	2	0	3	0	2	0	3	0	0	0
CO2	3	0	2	0	0	2	1	0	0	0	3	3
CO3	3	0	1	0	0	1	1	0	0	0	0	0
CO4	3	0	0	0	3	0	0	3	0	0	3	0
CO5	3	0	3	0	0	3	0	0	0	0	0	2

Course Code	:	**CSPC42**
Course Title	:	**Design and Analysis of Algorithms**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC32**
Course Type	:	**PC**

Course Objectives

- To understand the importance of algorithm
- To analyze the complexity of an algorithm in terms of time and space complexities
- To understand various problem solving techniques
- To learn about amortized analysis of algorithms
- To design and implement various programming paradigms and its complexity

Course Contents

UNIT I Introduction
Algorithms - Examples - Tournament method - Evaluating polynomial functions - pre-processing of coefficients - solving recurrence equations.

UNIT II Divide & Conquer and Greedy Approaches
Divide and Conquer method - Strassen's matrix multiplication - Greedy method - Huffman code - Minimum spanning trees - Dijkstra algorithm - Knapsack problem - Job sequencing with deadlines.

UNIT III Dynamic Programming Approaches
Dynamic Programming - Knapsack problem - Matrix Chain Multiplication - longest common subsequence Multistage graphs - All pair's shortest paths - Optimal binary search trees - Travelling salesman problem.

UNIT IV Amortization
Randomized Algorithms and Amortized Analysis - Las Vegas and Monte Carlo types - Randomized quick sort and its analysis - Min-Cut algorithm.

UNIT V NP Problems
NP-Hard and NP-complete problems - Basic concepts - Reducibility - Vertex cover-3 - CNF - clique - Hamiltonian cycle - TSP - Approximation algorithms - Vertex cover - TSP.

Course Outcomes
Upon completion of this course, the students will be able to:

- Analyze the time and space complexity for any algorithm
- Apply the design techniques of algorithm in solving real world problems
- Perform amortize analysis for any algorithm
- Design randomized and dynamic programming based algorithms
- Understand NP class of problems and propose approximation algorithms for the same

Text Book

1. T. Cormen, C. Lieserson, R. Rivest, C. Stein, "Introductions to Algorithms", Third Edition, Prentice-Hall/India, 2009.

Reference Books

1. M. Tenenbaum, Augestien, "Data Structures using C", Third Edition, Pearson Education, 2007.
2. Sartaj Sahni, "Data Structures, Algorithms and Applications in C++", Universities Press Pvt. Ltd., 2012.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	0	2	0	0	0	2	0	2	0
CO2	3	2	1	0	2	2	0	1	0	0	2	1
CO3	3	2	1	0	2	2	0	0	0	0	2	0
CO4	3	2	0	0	2	3	0	0	0	0	2	0
CO5	3	2	0	0	0	2	0	0	0	0	0	1

Course Code	:	**CSPC43**
Course Title	:	**Operating Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PC**

Course Objectives

- To provide knowledge about the services rendered by operating systems
- To explore the various scheduling policies and to provide solutions for critical section and deadlock problems
- To provide a detailed discussion of the various memory management techniques
- To discuss the various file-system design and implementation issues
- To discuss how the protection domains, help to achieve security in a system
- To explore the design and implementation issues of Distributed OS

Course Contents

UNIT I Introduction
Need for Operating Systems - Computer Systems - OS Operations - Abstract view of OS - Virtualization - Computing Environments - OS Services - OS Structures - System Calls - Building and Booting OS - Process - Threads - Multithreading.

UNIT II Process Management
Process Scheduling - Process Co-ordination - Synchronization - Semaphores - Monitors - Hardware Synchronization - Deadlocks - Methods for Handling Deadlocks.

UNIT III Memory Management
Memory Management Strategies - Contiguous and Non-Contiguous allocation - Virtual memory Management - Demand Paging - Page Placement and Replacement Policies.

UNIT IV File Management
File System - Basic concepts - File System design and Implementation - Case Study: Linux File Systems - Mass Storage Structure - Disk Scheduling - Disk Management - I/O Systems - System Protection and Security.

UNIT V Distributed Systems
Distributed Systems - Distributed operating systems - Distributed file systems - Distributed Synchronization - OS architecture - Case study on LINUX and Windows OS.

Course Outcomes

Upon completion of the course, the students will be able to:

- Comprehend the techniques used to implement the process manager
- Comprehend virtual memory abstractions in operating systems
- Design and develop file system and I/O system
- Apply various mechanisms in storage management
- Design and develop OS modules for Distributed Environment

Text Book

1. Silberschatz, Galvin, Gagne, “Operating System Concepts”, Tenth Edition, John Wiley and Sons, 2018.

Reference Books

1. William Stallings, “Operating Systems – Internals and Design Principles”, Eighth Edition, Pearson Publications, 2014.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Fourth Edition, Pearson Publications, 2014.
3. Dhananjay M. Dhamdhere, “Operating Systems - A Concept-Based Approach”, Third Edition, Tata McGraw Hill Education, 2012.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	3	0	3	0	2	0	0	0	2	0
CO2	3	0	3	0	0	0	2	0	0	0	0	1
CO3	3	0	3	0	0	0	2	0	0	0	0	1
CO4	3	0	1	0	0	0	3	0	0	0	0	2
CO5	3	2	0	0	3	0	0	0	2	1	3	0

Course Code	:	**CSLR41**
Course Title	:	**Algorithms Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PL**

Course Objectives

- To learn how to analyze the complexity of algorithms
- To compare and evaluate algorithms in terms of time and space complexity
- To program brute force, divide and conquer, decrease and conquer, transform and conquer, greedy, and dynamic techniques

Exercises

1. Estimating worst-case/average-case complexity of algorithms via programs.
2. Determining machine constants.
3. Programs involving some advanced data structures.
4. Implementing example problems.
5. Illustrating the different paradigms of algorithm design.
6. Solving miscellaneous problems e.g. problems in string manipulation, graph theory, optimization.

Course Outcomes

Upon completion of this course, the students will be able to:

- Solve and analyze general algorithms based on space and time complexity
- Implement and empirically compare fundamental algorithms and data structures to real-world problems
- Design, develop, and optimize algorithms in different paradigms
- Leaning implementation of advanced data structures
- Implement application specific data structure

Text Books

1. H. S. Wilf, “Algorithms and Complexity”, Prentice Hall.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, “Introduction to Algorithms”, Prentice Hall.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	0	0	2	0	0	2	0	0	0	2
CO2	3	2	1	0	2	2	0	0	2	0	2	0
CO3	3	0	1	0	2	2	0	0	0	0	0	2
CO4	3	1	0	0	2	0	0	0	0	0	0	0
CO5	3	1	0	0	2	3	0	0	0	0	2	0

Course Code	:	**CSLR42**
Course Title		**Operating Systems Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PL**

Course Objectives

- To understand the concept of Operating System
- To have insight knowledge on different system calls and Unix Utilities
- To experience the practical side of the functioning of various blocks in OS
- To design a real world application by considering process synchronization, Memory management

Exercises

1. Hands on Unix Commands.
2. Shell programming for file handling.
3. Shell Script programming using the commands grep, awk, and sed.
4. Programs on Multithread using Pthread.
5. Implementation of CPU scheduling algorithms.
6. Implementation of Synchronization problems using Semaphores, Message Queues and Shared Memory.
7. Implementation of Memory Management - Allocation, Placement and replacement Algorithms.
8. Implementation of various Disk scheduling algorithms.

Course Outcomes

Upon completion of the course, the students will be able to:

- Write program on shell script and Pthread
- Solve synchronization problems
- Compare and contrast various CPU scheduling algorithms, Memory allocation policy
- Differentiate the disk scheduling algorithms
- Learn memory management

Text Books

1. Silberschatz, Galvin, Gagne, "Operating System Concepts", Ninth Edition, John Wiley and Sons, 2013.
2. William Stallings, "Operating Systems – Internals and Design Principles", Eighth Edition, Pearson Publications, 2014.

Reference Books

1. Behrouz A. Forouzan, and Richard F. Gilberg, "UNIX and Shell Programming: A Textbook", Brooks/ Cole-Thomson Learning, 2003.
2. Andrew S. Tanenbaum, "Modern Operating Systems", Fourth Edition, Pearson Publications, 2014.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	1	0	2	0	0	0	0	0	0	0
CO2	3	0	0	0	2	0	0	0	2	0	2	0
CO3	3	2	1	0	0	0	0	0	0	0	0	0
CO4	3	2	0	0	0	0	0	0	0	0	0	0
CO5	3	2	0	0	2	0	0	0	0	0	0	0

V SEMESTER

Course Code	:	**CSPC51**
Course Title	:	**Computer Architecture**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**CSPC34**
Course Type	:	**PC**

Course Objectives

- To understand the concept of advanced pipelining techniques
- To understand the current state of art in memory system design
- To know the working principle of I/O devices
- To understand the memory management techniques

Course Contents

UNIT I
Introduction - Classes of computers - Defining Computer Architecture - Trends in Technology - Trends in Power and Energy in Integrated Circuits - Trends in Cost - Dependability - Measuring - Reporting and Summarizing Performance - Quantitative Principles of Computer Design.

UNIT II
Basic and Intermediate pipelining Concepts - The Major Hurdle of Pipelining - Pipeline Hazards - Pipelining Implementation - Implementation issues that makes Pipelining hard - Extending the MIPS Pipeline to Handle Multicycle Operations - The MIPS R4000 Pipeline.

UNIT III
Instruction-Level Parallelism: Concepts and Challenges - Basic Compiler Techniques for Exposing ILP - Reducing Branch Costs with Prediction - Overcoming Data Hazards with Dynamic Scheduling - Dynamic Scheduling - Hardware-Based Speculation - Exploiting ILP Using Multiple Issue and Static Scheduling - Exploiting ILP - Advanced Techniques for Instruction Delivery and Speculation - Studies of the Limitations of ILP.

UNIT IV
Vector Architecture - SIMD Instruction Set Extensions for Multimedia - Graphics Processing Units - Detecting and Enhancing Loop-Level Parallelism - Centralized Shared-Memory Architectures - Performance of Shared-Memory Multiprocessors - Distributed Shared Memory - Models of Memory Consistency - Multicore Processors and their Performance.

UNIT V
Review of Memory Hierarchy Design - Cache Performance - Basic Cache Optimizations - Virtual Memory - Protection and Examples of Virtual Memory - Advanced Optimizations of Cache Performance - Memory Technology and Optimizations - Protection: Virtual Memory and Virtual Machines - Crosscutting Issues: The Design of Memory Hierarchies - Case Studies / Lab Exercises.

Course Outcomes

Upon completion of the course, the students will be able to:

- Apply performance metrics to find the performance of systems
- Identify the program block that requires parallelism for any program
- Comprehend and differentiate various computer architectures and hardware
- Design algorithms for memory management techniques
- Analyse the performance of a system by applying various Cache memory optimization techniques

Text Books

1. David. A. Patterson, John L. Hennessy, “Computer Architecture: A Quantitative approach”, Fifth Edition, Elsevier, 2012.
2. K. Hwang, Naresh Jotwani, “Advanced Computer Architecture, Parallelism, Scalability, Programmability”, Second Edition, Tata McGraw Hill, 2010.

Reference Books

1. V. Carl Hamacher, Zvonko G. Varanesic, Safat G. Zaky, “Computer Organisation“, Sixth Edition, McGraw Hill Inc, 2012.
2. William Stallings “Computer Organization and Architecture”, Seventh Edition, Pearson Education, 2006.
3. Vincent P. Heuring, Harry F. Jordan, “Computer System Architecture”, Second Edition, Pearson Education, 2005.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	3	0	0	0	2	0	0	3	0	0
CO2	3	0	2	0	3	0	3	0	0	0	0	3
CO3	3	0	0	0	0	0	3	0	0	0	0	0
CO4	3	0	0	0	2	0	0	3	0	0	2	3
CO5	3	3	0	2	0	2	0	3	0	2	3	3

Course Code	:	**CSPC52**
Course Title	:	**Database Management Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PC**

Course Objectives

- To learn data models, conceptualize and depict a database system using ER diagram
- To understand the internal storage structures in a physical DB design
- To know the fundamental concepts of transaction processing techniques
- To understand the concept of Database Design in Normalization techniques
- To know the manipulation of SQL Queries

Course Contents

UNIT I Introduction
Purpose of Database System - Views of data - data models - database management system - three-schema architecture of DBMS - components of DBMS - E/R Model - Conceptual data modelling - motivation - entities - entity types - attributes - relationships - relationship types - E/R diagram notation - examples.

UNIT II Relational Model
Relational Data Model - Concept of relations - schema-instance distinction - keys - referential integrity and foreign keys - relational algebra operators - SQL - Introduction - data definition in SQL - table - key and foreign key definitions - update behaviours - Querying in SQL - notion of aggregation - aggregation functions groupby and having clauses - embedded SQL.

UNIT III Database Design
Dependencies and Normal forms - dependency theory - functional dependencies - Armstrong's axioms for FD's - closure of a set of FD's - minimal covers - definitions of 1NF - 2NF - 3NF and BCNF - decompositions and desirable properties of them - algorithms for 3NF and BCNF normalization - 4NF and 5NF.

UNIT IV Transactions
Transaction processing and Error recovery - concepts of transaction processing - ACID properties - concurrency control - locking based protocols for CC - error recovery and logging - undo - redo - undo-redo logging and recovery methods.

UNIT V Implementation Techniques
Data Storage and Indexes - file organizations - primary and secondary index structures - various index structures - hash-based dynamic hashing techniques - multi-level indexes - B+ trees.

Course Outcomes

Upon completion of this course, the students will be able to:

- Install, configure, and interact with a relational database management system
- Master the basics of SQL and construct queries using SQL
- Design and develop a large database with optimal query processing
- Develop efficient storage scheme of saving and retrieving Records and Files
- Design the database with normalization techniques

Text Books

1. Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Fifth Edition, Tata McGraw Hill, 2006.
2. J. Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

Reference Books

1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Fourth Edition, Pearson/Addision Wesley, 2007.
2. Raghu Ramakrishnan, “Database Management Systems”, Third Edition, McGraw Hill, 2003.
3. S. K. Singh, “Database Systems Concepts, Design and Applications”, First Edition, Pearson Education, 2006.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	2	2	0	0	0	0	0	0	0	0
CO2	2	2	0	0	0	0	0	0	0	0	0	0
CO3	2	0	3	3	2	3	0	1	2	0	2	1
CO4	2	2	2	2	2	3	0	1	2	0	2	1
CO5	0	0	0	2	1	1	1	1	1	1	2	1

Course Code	:	**CSPC53**
Course Title	:	**Computer Networks**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	-
Course Type	:	**PC**

Course Objectives

- To provide insight about fundamental concepts and reference models (OSI and TCP/IP) and its functionalists
- To gain comprehensive knowledge about the principles, protocols, and significance of Layers in OSI and TCP/IP
- To know the implementation of various protocols and cryptography techniques

Course Contents

UNIT I
Introduction to computer networks: Network Component and Categories - Topologies - Transmission Media - Reference Models: ISO/OSI Model and TCP/IP Model.

UNIT II
Physical Layer: Digital and Analog Signals - Periodic Analog Signals - Transmission Impairments - Digital data transmission techniques - Analog data transmission techniques - Multiplexing and Spread Spectrum.

UNIT III
Data Link Layer: Error Detection and Correction - Parity - LRC - CRC - Hamming Code - Flow Control and Error Control - Stop and wait - ARQ - Sliding window - HDLC - Multiple Access Protocols - CSMA - CSMA/CD and CSMA/CA - IEEE 802.3 Ethernet.

UNIT IV
Network Layer: Packet Switching and Datagram approach - IP Addressing methods - Subnetting - Routing - Distance Vector Routing - RIP - Link State Routing - OSPF - BGP - Multicast Routing - MOSPF - DVMRP - Broadcast Routing.

UNIT V
Transport Layer: Transport Services - UDP - TCP - Congestion Control - Quality of Services (QOS) - Application Layer: Domain Name Space (DNS) - Electronic Mail - WWW - Cryptography Techniques.

Course Outcomes

Upon completion of the course, the students will be able to:

- Gain insight about basic network theory and layered communication architectures
- Propose algorithms at the appropriate layer for any communication network task
- Provide solutions to various problems in network theory
- Conceptualize and design a network stack
- Assess the network service quality and applications of cryptography

Text Books

1. Andrew S. Tanenbaum, David J. Wetherall, “Computer Networks”, Fifth Edition, Prentice Hall, 2011.
2. Behrouz A. Foruzan, “Data Communication and Networking”, Fifth Edition, Science Engineering & Math Publications, 2013.

Reference Books

1. W. Stallings, “Data and Computer Communication”, Tenth Edition, Pearson Education, 2014.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	0	0	0	0	0	3	0	3	0	0
CO2	0	3	0	3	0	0	0	0	0	0	0	0
CO3	0	2	0	0	0	2	0	0	0	0	0	0
CO4	3	0	0	0	2	0	3	0	2	0	0	3
CO5	0	2	0	0	0	0	3	0	0	0	2	0

Course Code	:	**CSPC54**
Course Title	:	**Introduction to Artificial Intelligence and Machine learning**
Number of Credits	:	**3-0-2-4**
Pre-requisites (Course Code)	:	**CSPC11**
Course Type	:	**PC**

Course Objectives

- To learn the concepts of searching for AI problems
- To learn about agents and knowledge representation
- To understand the various factors involved in inferences
- To get introduced to fundamentals of machine learning
- To learn about the possibilities of Supervised and Unsupervised learning

Course Contents

UNIT I

AI - History of AI - Agents - Structure of Intelligent agents - Environments - Problem solving methods - Problem solving agents - Formulating problems - search strategies - Breadth-first - Uniform cost - Depth-first - Depth-limited - Bidirectional - Informed Search - Best-first Heuristic Functions - Memory bounded search - A* - SMA* - Iterative Improvement algorithms - Hill Climbing - Simulated annealing - Measure of performance and analysis of search algorithms.

Lab Component (Exercises similar to the following):

1. Heuristics and search strategy for Travelling salesperson problem.
2. Implement n-queens problem using Hill-climbing, simulated annealing, etc.

UNIT II

Game playing - Perfect Decisions - Imperfect Decisions - Alpha-beta pruning - Knowledge based agent - Wumpus World Environment - Propositional logic - agent for wumpus world - First order logic - syntax - semantics - extensions - Using First order logic - Representation change in the world - Goal based agents.

Lab Component (Exercises similar to the following):

1. Tic-tac-toe game simulation using search and heuristics.
2. Solve 3-SAT, 3-CNF algorithms using agents.
3. Describe the Sudoku game and represent the actions using First-order / Propositional logic.

UNIT III

Knowledge Base - Knowledge representation - Production based system - Frame based system - Inference - Backward chaining - Forward chaining.

Lab Component (Exercises similar to the following):

1. Sorting algorithms employing forward chaining.
2. Logical reasoning examples for E-commerce stores using forward/backward chaining.

UNIT IV

Learning from agents - inductive learning - Types of Machine learning - Supervised learning - learning decision trees - support vector machines - Neural and Belief networks - Perceptron - Multi-layer feed forward networks - Bayesian belief networks.

Lab Component (Exercises similar to the following):

1. Study of Machine learning tool.
2. Exercises on decision trees, SVM using the tool.

UNIT V

Unsupervised learning - K-means clustering - hierarchical clustering - Agglomerative and Divisive clustering - Fuzzy clustering.

Lab Component (Exercises similar to the following):

1. K-means clustering implementation using tool.
2. Agglomerative, divisive, fuzzy clustering using tool.

Course Outcomes

Upon completion of the course, the students will be able to:

- Suggest appropriate search strategies for any AI problem
- Design agents for any given problem
- Represent real world knowledge using first order or propositional logic
- Solve problems by appropriated using the supervised or unsupervised machine learning algorithms
- Suggest appropriate clustering algorithm for solving real-world problems

Text Books

1. Stuart Russel, Peter Norvig, "AI – A Modern Approach", Second Edition, Pearson Education, 2007.
2. Kevin Night, Elaine Rich, Nair B., "Artificial Intelligence (SIE)", McGraw Hill, 2008.

Reference Books

1. Vinod Chandra SS, Anand Hareendran S, "Artificial and Machine Learning", First Edition, PHI Learning, 2014.
2. Dan W. Patterson, "Introduction to AI and ES", Pearson Education, 2007
3. G. Luger, W. A. Sttubblefield, "Artificial Intelligence", Third Edition, Addison-Wesley Longman, 1998.
4. N. J. Nilson, "Principles of Artificial Intelligence", Narosa Publishing House, 1980.
5. Tom Mitchell, "Machine Learning", First Edition, Tata McGraw Hill India, 2017.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	0	3	0	0	0	3	0	0	0
CO2	3	0	3	2	0	3	0	0	0	0	0	0
CO3	2	2	3	0	0	0	0	0	0	0	0	0
CO4	3	0	3	0	0	0	0	0	0	0	0	0
CO5	3	3	3	0	0	0	0	0	0	0	0	0

Course Code	:	**CSLR51**
Course Title	:	**Database Management Systems Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PL**

Course Objectives

- To explore the features of a Database Management Systems
- To interface a database with front end tools
- To understand the internals of a database system
- To identify Structure Query Language statements used in creation and manipulation of Database
- To identify the methodology of conceptual modeling through Entity Relationship model

Exercises

1. Working with DDL, DML and DCL.
2. Inbuilt functions in RDBMS.
3. Nested Queries & Join Queries.
4. Set operators & Views in SQL.
5. Control structures.
6. Working with Procedures and Functions.
7. Triggers.
8. Dynamic & Embedded SQL.
9. Working with XML.
10. Forms & Reports.
11. Database Design and implementation (Mini Project).

Course Outcomes

Upon completion of this course, the students will be able to:

- Identify Structure Query Language statements used in creation and manipulation of Database
- Use databases for building client server applications
- Comprehend the internal working of a database system
- Design and develop a database using SQL and the mechanism in connecting with a Web based GUI
- Analyze and design a real database application

Text Books

1. Silberschatz, Henry F. Korth, S. Sudharshan, "Database System Concepts", Fifth Edition, Tata McGraw Hill, 2006.
2. C. J. Date, A. Kannan, S. Swamynathan, "An Introduction to Database Systems", Eighth Edition, Pearson Education, 2006.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	1	0	1	0	0	0	0	0	1	0
CO2	3	0	1	0	1	2	0	0	0	2	0	1
CO3	2	2	1	0	0	2	1	0	0	0	2	0
CO4	3	0	1	0	1	2	1	2	0	2	0	1
CO5	3	2	1	2	0	2	0	2	0	2	0	1

Course Code	:	**CSLR52**
Course Title	:	**Networks Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PL**

Course Objectives

- To create client and server applications using the "Sockets" API and the implementation of Data link layer protocol and TCP layer
- To conduct computer communication network simulations
- To have a hands on experience of computer network simulation and modelling techniques using NS-3 simulation software

Exercises

1. Exercises on Socket Programming using C and Java
2. Exercises using NS-3 Network Simulator
   a. Basics of Network Simulation
      - Introduction , Platform required to run network simulator, Backend Environment of Network Simulator, Agents and applications, Tracing
   b. Simulating a Local Area Network
      - Local Area Network, LAN Topologies, MAC Protocol, Taking turns, Ethernet, Ethernet Frame Structure, Ethernet Versions, Simulating a LAN using Network Simulator3
      - Implementation of various MAC protocols
      - Setting up of various network topologies
      - Measurement of routing protocols
   c. Measuring Network Performance
      - Network Performance Evaluation, Performance Evaluation Metrics, Parameters Affecting the Performance of Networks, Performance Evaluation Techniques, Network Performance Evaluation using NS-3
      - Setting up of network that carries various application protocols and analyzing the performances
3. Hands on experiments on Network equipments
   a. Switches, Routers
   b. Hardware firewall

Course Outcomes

Upon completion of the course, the students will be able to:

- Implement client-server applications using Sockets
- Invoke analytical studies of Computer Networks through network simulation
- Design a network using NS-3 toolkit and its importance in designing a real network
- Measure and analyze the network parameters for a high throughput network
- Practice experiments on Network Equipments

Text Books

1. W. Richard Stevens, "UNIX Network Programming – Networking APIs: Sockets and XTI", Vol. 1, Second Edition, Prentice Hall, 1998.
2. Eitan Altman, Tania Jimenez, "NS Simulator for Beginners", Morgan & Claypool Publishers, 2011.

Reference Book

1. Jack L. Burbank, "An Introduction to Network Simulator 3", First Edition, Wiley-Blackwell, 2015.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	0	1	1	2	2	0	0	0	0	0
CO2	1	0	1	2	0	0	2	0	0	0	3	0
CO3	0	1	2	0	3	3	0	0	2	0	0	3
CO4	0	1	1	0	0	2	0	3	0	3	0	0
CO5	0	0	1	0	0	2	3	0	0	3	0	0

VI SEMESTER

Course Code	:	**CSPC61**
Course Title	:	**Embedded Systems Architectures**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC51**
Course Type	:	**PC**

Course Objectives

- To understand basics of Embedded Systems Architecture
- To understand the intricacies of Embedded programming

Course Contents

Unit I Introduction to Embedded System
Introduction - A Systems Engineering Approach to Embedded System Design - Architecture - Importance - System model - Programming Languages and Examples - Standards and Networking - Multiple standards-based device.

Unit II Embedded Hardware: Hardware Building Blocks
The Embedded Board and the Von-Neumann Model - Basic Hardware Materials: Embedded Processors - ISA Architecture Models - Internal processor design - Processor Performance - Memory - Board I/O - Board Buses - Component Interfacing.

Unit III Embedded Software: Device Drivers
Interrupt - Handling - Memory Device Drivers - On-board Bus Device Drivers - Examples - Embedded Operating Systems - Process - Multitasking and Process Management - I/O and File System Management.

Unit IV OS for Embedded Systems
Process - Multitasking and Process Management - POSIX - OS Performance Guidelines - selecting right OS's and Board Support Packages (BSPs) - Middleware and Application Software - Development Tools for Embedded System - Embedded C programming.

Unit V Design, Development and Case studies
Creating an Embedded System Architecture - Implementation and Testing - Implementing the Design - Quality Assurance and Testing of the Design - Debugging - System Level Performance Analysis - Maintaining the Embedded System - Embedded GPU Design - Embedded Computing System on FPGAs - Hardware-Software Co-design - Embedded Systems Security - Typical Case Studies: Automotive Driver Assistance - Mobile Agents for Embedded System.

Course Outcomes

- Ability to comprehend the architecture of Embedded systems
- Ability to design embedded systems for simple tasks
- Ability to design and develop programs for specific embedded applications
- Understanding operating systems for embedded systems
- Understand about life cycle of embedded design and its testing.

Text Books

1. Tammy Noergaard, "Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers", Second Edition, Elsevier Embedded Technology Series, Newnes Publication, 2012.
2. Krzysztof Iniewski, "Embedded Systems: Hardware, Design, and Implementation", Wiley & Sons, Inc. Edited, 2013.
3. Raj Kamal, "Embedded Systems: Architecture, Programming and Design", Third Edition, McGraw Hill Education (India), 2014.

Reference Books

1. Julio Sanchez, Maria P. Canton, “Embedded Systems Circuits and Programming”, Taylor and Francis, 2012.
2. J. Staunstrup, Wayne Wolf, “Hardware/Software Co-Design: Principles and Practice”, Prentice Hall.
3. Michael Barr, and Anthony Massa, “Programming Embedded Systems: With C and GNU Development Tools”, Second Edition, O’Reilly, 2007.
4. Wayne Wolf, “Computers as Components - Principles of Embedded Computer System Design”, Morgan Third Edition, Kaufmann Publishers, 2012.
5. Sriram V Iyer, Pankaj Gupta, “Embedded Real Time System Programming”, Tata McGraw Hill, 2004.
6. Jack Ganssle, “Embedded Systems: World Class Designs”, Elsevier, 2008.
7. Kiyofumi Tanaka, “Embedded Systems: High Performance Systems, Applications and Projects”, Intech Publication, 2012.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	1	0	0	0	0	2	2	0	0	2	2
CO2	3	0	1	1	3	3	2	0	0	0	0	1
CO3	3	0	1	1	0	0	0	0	0	2	0	1
CO4	0	2	0	0	1	0	0	0	0	1	0	0
CO5	0	2	0	0	1	0	0	0	0	1	0	0

Course Code	:	**CSPC62**
Course Title	:	**Compiler Design**
Number of Credits	:	**3-0-2-4**
Pre-requisites (Course Code)	:	**CSPC41**
Course Type	:	**PC**

Course Objectives

- To introduce the major concept areas in compiler design and know the various phases of the compiler
- To understand the various parsing algorithms and comparison of the same
- To provide practical programming skills necessary for designing a compiler
- To gain knowledge about the various code generation principles
- To understand the necessity for code optimization

Course Contents

UNIT I Introduction to Compilation

Compilers - Analysis of the source program - Phases of a compiler - Cousins of the Compiler - Grouping of Phases - Compiler construction tools - Lexical Analysis - Role of Lexical Analyzer - Input Buffering - Specification of Tokens.

Lab Component: Tutorial on LEX / FLEX tool, Tokenization exercises using LEX.

UNIT II Syntax Analysis

Role of the parser - Writing Grammars - Context-Free Grammars - Top Down parsing - Recursive Descent Parsing - Predictive Parsing - Bottom-up parsing - Shift Reduce Parsing - Operator Precedent Parsing - LR Parsers - SLR Parser - Canonical LR Parser - LALR Parser.

Lab Component: Tutorial on YACC tool, Parsing exercises using YACC tool.

UNIT III Intermediate Code Generation

Intermediate languages - Declarations - Assignment Statements - Boolean Expressions - Case Statements - Back patching - Procedure calls.

Lab Component: A sample language like C-lite is to be chosen. Intermediate code generation exercises for assignment statements, loops, conditional statements using LEX/YACC.

UNIT IV Code Optimization and Run Time Environments

Introduction - Principal Sources of Optimization - Optimization of basic Blocks - DAG representation of Basic Blocks - Introduction to Global Data Flow Analysis - Runtime Environments - Source Language issues - Storage Organization - Storage Allocation strategies - Access to non-local names - Parameter Passing - Error detection and recovery.

Lab Component: Local optimization to be implemented using LEX/YACC for the sample language.

UNIT V Code Generation

Issues in the design of code generator - The target machine - Runtime Storage management - Basic Blocks and Flow Graphs - Next-use Information - A simple Code generator - DAG based code generation - Peephole Optimization.

Lab Component: DAG construction, Simple Code Generator implementation, DAG based code generation using LEX/YACC for the sample language.

Course Outcomes

Upon completion of this course, the students will be able to:

- Apply the knowledge of LEX & YACC tool to develop a scanner and parser
- Design and develop software system for backend of the compiler
- Suggest the necessity for appropriate code optimization techniques
- Conclude the appropriate code generator algorithm for a given source language
- Design a compiler for any programming language

Text Books

1. Alfred V. Aho, Jeffrey D Ullman, “Compilers: Principles, Techniques and Tools”, Pearson Education Asia, 2012.
2. Jean Paul Tremblay, Paul G Serenson, “The Theory and Practice of Compiler Writing”, BS Publications, 2005.
3. Dhamdhere, D. M., “Compiler Construction Principles and Practice”, Second Edition, Macmillan India Ltd., New Delhi, 2008.

Reference Books

1. Allen I. Holub, “Compiler Design in C”, Prentice Hall of India, 2003.
2. C. N. Fischer, R. J. LeBlanc, “Crafting a compiler with C”, Benjamin Cummings, 2003.
3. Henk Alblas, Albert Nymeyer, “Practice and Principles of Compiler Building with C”, PHI, 2001.
4. Kenneth C. Louden, “Compiler Construction: Principles and Practice”, Thompson Learning, 2003.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	1	0	0	3	0	0	0	3	0	2
CO2	1	0	2	3	0	0	0	0	0	3	0	0
CO3	2	0	2	0	0	0	0	2	0	3	1	2
CO4	0	1	0	2	0	3	0	2	0	0	2	2
CO5	2	1	0	0	0	2	3	0	0	3	1	1

Course Code	:	**CSPC63**
Course Title	:	**Principles of Cryptography**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	-
Course Type	:	**PC**

Course Objectives

- To gain knowledge about the mathematics of the cryptographic algorithms
- To get an insight into the working of different existing cryptographic algorithms
- To learn about key exchange protocols and attacks on such protocols
- To introduce the fundamental concepts of hash functions and digital signatures
- To learn how to use cryptographic algorithms in security

Course Contents

UNITI Mathematical Foundations

Number Theory: Fermat's theorem - Cauchy 's theorem - Chinese remainder theorem - Primality testing algorithm - Euclid's algorithm for integers - quadratic residues - Legendre symbol - Jacobi symbol.*

UNIT II Classical Cryptosystems

Cryptography and cryptanalysis - Classical Cryptography - different type of attack: CMA - CPA - CCA - Shannon perfect secrecy - OTP - Pseudo random bit generators - stream ciphers and RC4.*

UNIT III Symmetric Key Ciphers

Block ciphers: Modes of operation - DES and its variants - finite fields (2^n) - AES - linear and differential cryptanalysis.*

UNIT IV Asymmetric Key Ciphers

One-way function - trapdoor one-way function - Public key cryptography - RSA cryptosystem - Diffie-Hellman key exchange algorithm - ElGamal Cryptosystem.*

UNIT V Message Authentication

Cryptographic hash functions - secure hash algorithm - Message authentication - digital signature - RSA digital signature.*

*Programming assignments are mandatory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Understand the basic concepts of symmetric cryptosystem, public key cryptosystem and digital signature scheme
- Reason about the security of cryptographic algorithms
- Evaluate the security of a protocol based on security metrics
- Justify the usage of security principles and digital signatures for any application
- Ability to break the cryptosystem that is secure

Text Book

1. Stinson. D., “Cryptography: Theory and Practice”, Third Edition, Chapman & Hall/CRC, 2012.

Reference Books

1. W. Mao, “Modern Cryptography: Theory & Practice”, Pearson Education, 2010.
2. W. Stallings, “Cryptography and Network Security Principles and practice”, Fifth Edition, Pearson Education Asia, 2013.
3. Behrouz A. Forouzan, Debdeep Mukhopadhyay, “Cryptography and Network Security”, Second Edition, Tata McGraw Hill, 2013.
4. Thomas Koshy, “Elementary Number Theory with Applications”, Elsevier India, 2005.
5. *[Online Course]* Course on Cryptography by Dan Boneh.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	3	0	0	3	2	0	0	0	0	0
CO2	3	0	3	3	3	3	0	0	0	0	0	0
CO3	3	0	3	0	3	3	0	0	0	0	0	0
CO4	0	1	0	3	1	0	1	0	0	2	1	2
CO5	0	0	1	0	0	2	0	3	1	0	0	0

Course Code	:	**CSLR61**
Course Title	:	**Embedded Systems Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PL**

Course Objectives

- To introduce embedded systems design tools
- To write programs to interface the memory and I/Os with processors
- To program and test the ARM processor based circuits and their interfaces
- To implement the basic building blocks of a microcontroller including counters, I/O techniques and requirements, A/D conversion

Exercises

1. To interface LED and perform the flashing of LEDs with ARM processor using mbed LPC 1768.
2. To create a waveforms using ARM processor using mbed LPC 1768 and display it using CRO.
   a. Triangular waveform
   b. Square waveform
   c. Saw-tooth waveform
3. To interface the DC motor with ARM processor using mbed LPC 1768 and perform the speed control.
4. To write and read data from EEPROM interfaced with ARM processor.
5. To interface LCD with ARM processor and display the text.
6. To implement SISO and PISO using Zybo board.
7. To implement SIPO and PIPO using Zybo board.
8. To implement 3-bit Counters in Zybo board.
   a. Ring Counter
   b. Johnson counter
9. To interface temperature sensor with Raspberry Pi.
10. To interface humidity sensor with Raspberry Pi.
11. To interface RFID sensor with Raspberry Pi.
12. To interface convert Analog signal in digital form using 3 – bit ADC in Zedboard.
13. To interface and convert Digital to Analog using 3 – bit DAC in Zedboard.
14. To interface external memory with Zedboard.
15. To interface monitor and external source with Zedboard.

Course Outcomes

Upon completion of this course, the students will be able to:

- Assemble and troubleshoot hardware devices
- Write programs for interfacing keyboard, display, motor and sensor
- Design and program an embedded system at the basic level
- Write programs in ARM for a specific Application
- Formulate a mini project using embedded systems

Text Books

1. Yu-Cheng Liu, Glenn A. Gibson, “Microcomputer Systems: The 8086/8088 Family-Architecture, Programming and Design”, Second Edition, Prentice Hall of India, 2009.
2. William Hohl, “ARM Assembly Language Fundamentals and Techniques”, 2012.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	2	3	1	0	0	0	0	0	0	1
CO2	3	3	3	3	1	1	0	0	0	1	0	0
CO3	3	0	1	1	1	0	0	0	0	1	0	0
CO4	3	3	3	2	1	1	0	0	0	0	0	0
CO5	3	3	3	3	3	3	1	0	3	3	3	1

Course Code	:	**CSLR62**
Course Title	:	**App Development Laboratory**
Number of Credits	:	**0-0-3-2**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PL**

Course Objectives

- To be familiar with Web page design using HTML/XML and style sheets
- To learn to write Client Server applications
- To be familiar with the PHP programming
- To be exposed to creating applications with AJAX
- Know the components and structure of mobile application development frameworks for Android and windows OS based mobiles
- Learn the basic and important design concepts and issues of development of mobile applications

Exercises

A. Web Applications

1. Create a web page for user registration using HTML, CSS and validate the details using Javascript.
2. Write programs in Java using Servlets: (i) To invoke servlets from HTML forms; (ii) Session tracking using hidden form fields and Session tracking for a hit count.
3. Create three-tier applications using JSP for conducting on-line examination for displaying student mark list. Assume that student information is available in a database which has been stored in a database server.
4. Create a database with user information and books information and create a webpage in which books catalogue should be dynamically loaded from the database using AJAX.
5. Create and save an XML document at the server, which contains 10 users Information. Write a Program, which takes user Id as an input and returns the User details by taking the user information from the XML document.
6. Develop email verification application using PHP.

B. Mobile Applications

1. Design restaurant data entry form using Table Layout and show different events using activity class.
2. Write a program to capture image using built in camera and store it in database.
3. Develop a banking application that registers the user by verifying OTP.
4. Develop a native application that uses GPS location information and convert into speech.
5. Write a program to call a number.

Course Outcomes

Upon completion of this course, the students will be able to:

- Construct Web pages using HTML/XML and style sheets
- Build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms
- Develop Web application which makes use of PHP programming
- Construct web applications using AJAX
- Design and Implement various mobile applications using emulators
- Deploy applications to hand-held devices

Text Books

1. Reto Meier, “Professional Android 4 Application Development”, Wrox, 2012.
2. Matt Gifford, “PhoneGap Mobile Application Development Cookbook”, Packt, 2012.
3. Adrian Kosmaczewski, “Mobile JavaScript Application Development”, O’Reilly, 2012.

Mapping of Course Outcomes with Programme Outcomes

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	3	2	3	3	2	3	2	3	3	0
CO2	3	2	3	3	3	3	0	3	0	3	3	2
CO3	3	3	3	2	3	3	0	3	0	3	0	2
CO4	3	2	3	2	2	3	0	3	2	3	2	2
CO5	3	2	3	3	2	3	2	0	3	3	2	3

Course Code	:	**HSIR14**
Course Title	:	**Professional Ethics**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**GIR**

Course Objectives

- Identify the core values that shape the ethical behavior of an engineer
- To create an awareness on professional ethics and human values
- To introduce loyalty, moral and social values
- To develop as a competent and trust worthy professionals
- To enable the students to appreciate the rights of others

Course Contents

UNIT I Human Values

Morals - Values and Ethics - Integrity - Work Ethic - Service Learning - Civic Virtue - Respect for others - Living Peacefully - Caring - Sharing - Honesty - Courage - Valuing time - Co-operation - Commitment - Empathy - Self-confidence - Character - Spirituality - The role of engineers in modern society - Social expectations.

UNIT II Engineering Ethics

Sense of 'Engineering Ethics' - Variety of moral issued - types of inquiry - moral dilemmas - moral autonomy - Kohlberg's theory - Gilligan's theory - Consensus and Controversy - Models of Professional Roles & Professionalism - theories about right action - Self-interest - customs and religion - uses of ethical theories.

UNIT III Engineering as Social Experimentation

Engineering as experimentation - engineers as responsible experimenters - Research ethics - Codes of ethics - Industrial Standard - Balanced outlook on law - the challenger case study.

UNIT IV Safety, Responsibilities and Rights

Safety and risk - assessment of safety and risk - Riysis - Risk benefit analysis and reducing risk - Govt. Regulator's approach to risks - the three mile island and Chernobyl case studies & Bhopal Threat of Nuclear Power - depletion of ozone - greenery effects - Collegiality and loyalty - respect for authority - collective bargaining - Confidentiality - conflicts of interest - occupation crime - professional rights - employees' rights - Intellectual Property Rights (IPR) - discrimination.

UNIT V Global Issues

Multinational corporations - Business ethics - Environmental ethics - computer ethics - Role in Technological Development - Weapons development - engineers as managers - consulting engineers - engineers as expert - witnesses and advisors - Honesty - Leadership - sample code of conduct ethics like ASME - ASCE - IEEE - Institution of Engineers (India) - Indian Institute of Materials Management - Institution of Electronics and Telecommunication Engineers (IETE India).

Course Outcomes

Upon completion of this course, the students will be able to:

- Understood the core values that shape the ethical behavior of an engineer
- Expose awareness on professional ethics and human values
- Understood the importance of research ethics and moral issues while conducting social experimentation
- Demonstrate universally accepted ethical standards
- Gain adequate knowledge about moral leadership and code of conduct

Text Books

1. Mika Martin, Roland Scinger, “Ethics in Engineeering”, Pearson Education/Prentice Hall, New York, 1996.
2. Govindarajan M., Natarajan S., Senthil Kumar V. S., “Engineering Ethics”, Prentice Hall of India, New Delhi, 2004.

Reference Books

1. Charles D. Fleddermann, “Ethics in Engineering”, Pearson Education/Prentice Hall, New Jersey, 2004 *(Indian Reprint)*.
2. Charles E. Harris, Michael S. Protchard, Michael J. Rabins, “Engineering Ethics – Concept and Cases”, Wadsworth Thompson Learning, United States, 2000 *(Indian Reprint now available)*.
3. Arthur A. Thompson et al., “Strategic Management: Concepts and Cases”, McGraw-Hill Higher Education, 2000.
4. John R. Boatright, “Ethics and Conduct of Business”, Pearson Education, New Delhi, 2003.
5. Edmund G. Seebauer, Robert L. Barry, “Fundamentals of Ethics for Scientists and Engineers”, Oxford University Press, Oxford, 2001.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	0	0	0	0	0	3	0	1	1	0
CO2	0	0	0	0	0	0	0	3	0	3	3	0
CO3	0	0	0	0	0	0	0	3	0	2	3	0
CO4	0	0	0	0	0	0	0	3	0	3	3	0
CO5	0	0	0	0	0	0	0	3	0	2	3	0

PROGRAMME ELECTIVES

III SEMESTER

Course Code	:	**CSPE31**
Course Title	:	**Linear Algebra and Applications**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand linear system of equations
- To gain knowledge about vector spaces and the operations performed on it
- To understand orthogonality principle of vectors and how it is useful in solving problems
- To acquire knowledge about properties of determinants
- To have an idea about eigen value and eigen vectors

Course Contents

UNIT I Matrices and Gaussian Elimination

Introduction and solving a linear system of equations: Introduction - Geometry of linear Equations - An Example of Gaussian Elimination - Matrix Notation and Matrix Multiplication - Triangular Factors and Row Exchanges - Inverses and Transposes - Finite difference matrices: tridiagonal and their LU decomposition.

UNIT II Orthogonality

Orthogonality of vectors - subspaces - notion of orthogonal compliment of a subspace - orthogonality relations between the four fundamental subspaces of a matrix - Solutions to least square error problems - connection to pseudo-inverse Projection onto a vector space as a matrix operation - projection onto a line - Minimum norm solution in the under-determined case - connection to pseudo-inverse Orthogonal vector and matrices - Gram-Schmidt process of orthonormalization - QR decomposition of a matrix - Hilbert spaces - function spaces and the concept of orthogonality in these spaces.

UNIT III Determinants

Properties of determinants - Formula of determinants - Applications of determinants - Geometrical interpretation of determinants - determinant of the Jacobian Permutations and Cofactors.

UNIT IV Eigen value and Eigen Vectors

Definition and a few properties of the matrix eigen value problem - Algebraic and geometric multiplicity of an eigen value - some properties - Proof regarding multiplicity - Diagonalization of a matrix (also called its eigen decomposition) - its use to compute powers of a matrix - Cayley-Hamilton theorem - Powers of a matrix: application based on Fibonacci numbers - Hermitian matrices and their properties - Spectral theorem - Unitary matrices - change of basis and similarity transforms - On the Schur decomposition of a matrix.

UNIT V Applications

Graphs and Networks - Linear programming - Markov matrices - Linear programming - fourier series.

Course Outcomes

Upon completion of this course, the students will be able to:

- Solve linear system of equations with LU decomposition
- Perform linear transformation with matrices
- Apply orthogonality relations between the subspaces of a matrix
- Understand the properties of determinants
- Use eigen value and eigen vectors to compute powers of a matrix

Text Book

1. Gilbert Strang , “Linear Algebra and its Applications”, Fourth Edition, GS.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	0	0	0	0	0	0	0	2
CO2	3	3	2	2	0	0	0	0	0	0	0	2
CO3	3	3	2	2	0	0	0	0	0	0	0	2
CO4	3	3	2	2	0	0	0	0	0	0	0	2
CO5	3	3	2	2	0	0	0	0	0	0	0	2

Course Code	:	**CSPE32**
Course Title	:	**Combinatorics and Graph Theory**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC11**
Course Type	:	

Course Objectives

- To introduce basic concepts of combinatorics and graph theory
- To study graphs, trees and networks
- To discuss Euler formula, Hamilton paths, planar graphs and coloring problem
- To practice useful algorithms on networks such as shortest path algorithm, minimal spanning tree algorithm and min-flow max-cut algorithm

Course Contents

UNIT I
Introduction to combinatorics - permutation of multisets - Combinations of Multisets - distribution of distinct objects into distinct cells - distribution of non-distinct objects into distinct cells - Shamire secret sharing - Catalan number - Principle of inclusion and exclusion - Derangement.

UNIT II
Generating functions - Partitions of integer - Ferrer graph - Solving recurrence relations using generating functions - Generating permutations and combinations - Pigeonhole principle: simple and strong Form - A THEOREM OFRAMSEY.

UNIT III
Graph - simple graph - graph isomorphism - incidence and adjacency matrices - Haveli-Hakimi criterion - Subgraphs Tree - minimum spanning tree - Kruskal - Prims algorithm - Caleys' formula - Kirchoff-Matrix-tree Theorem - Fundamental circuits - Algorithms for fundamental circuits - Cut-sets and Cut-vertices - fundamental cut-sets.

UNIT IV
Eular graph - Fleury's algorithm - Hamiltonian graph - Planar and Dual Graphs - Kuratowski's graphs Coloring - Greedy coloring algorithm - chromatic polynomial.

UNIT V
Mycielski's theorem - Matching halls marriage problem - Independent set - Dominating set - Vertex cover - clique - approximation algorithms.

Course Outcomes

Upon completion of the course, the students will be able to:

- Comprehend the fundamentals of combinatorics and apply combinatorial ideas in mathematical arguments in analysis o algorithms, queuing theory,etc.
- Comprehend graph theory fundamentals and tackle problems in dynamic programming, network flows,etc.
- Learn role of complexity in combinatorics and graph theory
- Design and develop real time application using graph theory
- Construct and communicate proofs of theorems

Text Books

1. Ralph P. Grimaldi, “Discrete and Combinatorial Mathematics”, Fifth Edition, PHI/Pearson Education, 2004.
2. G. Chartrand, P. Zhang, “Introduction to Graph Theory”, McGraw-Hill, 2006.

Reference Books

1. Kenneth H. Rosen, “Discrete Mathematics and its Applications”, Seventh Edition, McGraw Hill, 2012.
2. John Harris, Jeffry L. Hirst, Michael Mossinghoff, “Combinatorics and Graph Theory”, Second Edition, Springer Science & Business Media, 2008.
3. J. H. Van Lint, R. M. Wilson, “A Course in Combinatorics”, Second Edition, Cambridge Univ. Press, 2001.
4. Dr. D. S. Chandrasekharaiah, “Graph Theory and Combinatorics”, Prism, 2005.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	0	0	2	0	0	0	0	0	0
CO2	1	3	2	2	0	0	0	0	0	0	0	3
CO3	1	0	2	0	2	3	0	0	0	0	2	0
CO4	3	0	3	0	2	2	0	0	0	0	2	2
CO5	0	2	0	0	0	0	0	1	0	3	2	0

IV SEMESTER

Course Code	:	**CSPE41**
Course Title	:	**Software Engineering**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand the Software Engineering Practice
- To understand the Software Engineering Process Models
- To understand Design Engineering, Web applications
- To gain knowledge of the software testing
- To understand Software Project Management

Course Contents

UNIT I
Introduction: Role of Software Engineer - Software Components - Software Characteristics - Software Crisis - Software Engineering Processes - Similarity and Differences from Conventional Engineering Processes - Quality Attributes.
Assessment: How Software Engineering Changes? Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models, Choosing a social relevant problem, Summary Team Report.

UNIT II
Requirement Engineering Process: Elicitation - Analysis - Documentation - Review and Management of User Needs - Feasibility Study - Information Modeling - Data Flow Diagrams - Entity Relationship Diagrams - Designing the architecture.
Assessment: Impact of Requirement Engineering in their problem, Decision Tables, SRS Document, IEEE Standards for SRS, Architectural design, component level design, user interface design, WebApp Design, Submission of SRS Document for Team Project.

UNIT III
Quality concepts - Review techniques - Software Quality Assurance (SQA): Verification and Validation - SQA Plans - Software Quality Frameworks.
Assessment: Framing SQA Plan, ISO 9000 Models, SEI-CMM Model and their relevance to project Management, other emerging models like People CMM.

UNIT IV
Testing Objectives - Unit Testing - Integration Testing - Acceptance Testing - Regression Testing - Testing for Functionality and Testing for Performance - Top-Down and Bottom-Up Testing - Software Testing Strategies - Strategies: Test Drivers and Test Stubs - Structural Testing (White Box Testing) - Functional Testing (Black Box Testing) - Testing conventional applications - object oriented applications - Web applications - Formal modeling and verification - Software configuration management - Product metrics.
Assessment: Team Analysis in Metrics Calculation.

UNIT V
Project Management Concepts - Process and Project Metrics - Estimation for Software projects - Project Scheduling - Risk Management - Maintenance and Re-engineering.
Assessment: Preparation of Risk mitigation plan.

Course Outcomes

Upon completion of the course, the students will be able to:

- Assess each module given the overall Software engineering practice
- Enhance the software project management skills
- Comprehend the systematic methodologies involved in SE
- Design and develop a software product in accordance with SE principles
- Design risk mitigation plans for software products.

Text Books

1. R. S. Pressman, "Software Engineering: A Practitioners Approach", Seventh Edition, McGraw Hill, 2010.
2. Rajib Mall, "Fundamentals of Software Engineering", Third Edition, PHI Publication, 2009.
3. Pankaj Jalote, "Software Project Management in Practice", Pearson Education, New Delhi, 2002.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	2	0	0	2	0	0	3	0	2	3	0
CO2	0	0	0	0	2	0	0	3	3	2	2	0
CO3	0	1	0	0	2	0	0	2	2	1	2	0
CO4	2	0	2	0	3	1	0	1	0	1	1	0
CO5	0	0	0	2	2	1	0	0	3	3	3	0

Course Code	:	**CSPE42**
Course Title	:	**Design Thinking**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand processes that enhances innovation activities
- To develop capabilities to identify problems/issues/needs
- To develop sound hypotheses, collect and analyze appropriate data
- To translate broadly defined opportunities into actionable innovation possibilities

Course Contents

UNIT I

Design Thinking - Introduction - What - How - Why - Design Process - Four Questions - Ten Tools - Identify an Opportunity - Scope your opportunity - Draft your design brief.

UNIT II

Three visualizations - Visualization basics - Journey mapping - Value Chain analysis - Mind mapping.

UNIT III

Design Criteria - Design thinking brainstorming - Concepts development - develop concepts - napkin pitches.

UNIT IV

Assumption testing - Rapid Prototyping - Surface Key assumptions - make prototypes.

UNIT V

Customer co-creation - learning launch - Feedback from stake holders - Design the on-ramp - Case study.

Course Outcomes

Upon completion of the course, the students will be able to:

- Convert real-life problems into methodical problems
- Apply various visualization principles for problem and solution representation
- Design solutions by applying an integrated approach to design thinking
- Justify and prototype solutions to problems
- Understanding customer feedback

Text Books

1. Jeanne Liedtka, Tim Ogilvie, Rachel Brozenske, “The Designing for Growth Field Book: A Step-by Step Project Guide”, New York: Columbia University Press, 2014.
2. Jeanne Liedtka, Tim Ogilvie, “Designing for Growth: A Design Thinking Tool Kit for Managers”, New York: Columbia University Press, 2011.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	1	0	2	0	0	0	2	0	0	0
CO2	0	0	0	0	2	3	0	2	0	1	2	0
CO3	3	0	1	0	2	0	0	0	2	0	0	2
CO4	3	0	1	0	0	0	0	2	0	3	3	0
CO5	0	0	0	0	0	0	0	0	0	3	0	0

Course Code	:	**CSPE43**
Course Title	:	**Advanced Data Structures and Algorithms**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC31**
Course Type	:	**PE**

Course Objectives

- To explore various heaps
- To study the need for various balanced and multimedia tree structures
- To study about geometric algorithms and their applications
- To understand the approximation algorithms for the various NP complete problems
- To study about the various string matching algorithms

Course Contents

UNIT I
Binomial Heaps - Deaps - Leftist Heaps - Fibonacci Heaps – Operations – insert - delete - meld - find min.

UNIT II
Splay Trees - Point trees - Quad trees - K-d trees - TV-trees - Segment trees.

UNIT III
Dynamic programming - Optimal Binary search trees - TSP - Graph coloring - Knapsack problem - Backtracking algorithms - N-queens - Hamiltonian cycle - Graph coloring - Branch and bound method – Knapsack - TSP.

UNIT IV
Number-theoretic algorithms - FFT - String matching algorithms - KMP - Rabin-Karp - Boyer Moore algorithms.

UNIT V
Computational Geometry - convex hull - NP Complete problems - Reducibility - Vertex cover–clique - Hamiltonian cycle - TSP.

Course Outcomes

Upon completion of this course, the students will be able to:

- Apply the appropriate heap data structure for solving real-world problems
- Use special tree data structures for a given real-world problem
- Decide on appropriate string matching algorithms for solving practical problems
- Appreciate the backtracking and branch and bound technique to solving NP problems
- Analyse geometric problems and NP-complete problems and demonstrate the impact of reducibility on the real time problems.

Text Books

1. H. S. Wilf, “Algorithms and Complexity”, Prentice Hall.
2. T. H. Cormen, C. E. Leiserson, R. L. Rivest, “Introduction to Algorithms”, Third Edition, Prentice Hall, 2012.
3. Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, “Fundamentals of Computer Algorithms”, Second Edition, Universities Press, Reprint 2017.
4. Subrahmanian, “Principles of Multimedia Database systems”, First Edition, Elsevier, 2008.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	1	2	2	0	0	0	0	2	0
CO2	1	1	2	2	0	3	0	0	3	0	3	0
CO3	1	1	1	2	2	3	0	0	3	0	3	0
CO4	0	2	1	0	2	0	0	3	0	0	3	0
CO5	1	1	2	0	0	3	3	0	0	0	0	0

Course Code	:	**CSPE44**
Course Title	:	**Computer Graphics**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand the basics of various input and output devices used in computer graphics
- To learn the basic to build 2D and 3D objects
- To understand 2D and 3D transformations and Viewing
- To provide comprehensive knowledge on hidden surface detection
- To be familiar with Animation, Morphing

Course Contents

UNIT I

Introduction: Basics and applications of Computer Graphics - Graphics software and standards - Interaction (sample- and event-driven) and Graphics user Interface (GUI) features.

Display Systems: Raster and Random displays - CRT basics - Flat panel displays - 3D display systems - Hardcopy devices–Printers and Plotters - Various File formats and Colour models.

UNIT II

Output Primitives: Point - Line - Circle – Ellipse - Scan conversion algorithms for primitives - Fill area primitives – scan - line - polygon filling - inside-outside test - boundary and flood fill - character generation - line attributes – area - fill attributes - character attributers.

OpenGL primitives: Functions - pipeline - sample programs for drawing 2-D and 3-D objects; event handling and view manipulation.

UNIT III

2D Transformations and Viewing: Rotation - Translation - Scale - Reflection and Shear Transform - Matrix representation - homogeneous co-ordinates - composite transformations - Clipping algorithms for point - line and polygon - Text.

UNIT IV

3D object representation: 3D display methods - polygon surfaces - tables - equations - meshes.

Curves and Surfaces: curved lines and surfaces - quadric surfaces - spline representation - cubic spline interpolation methods - Bezier curves and surfaces - B-spline curves and surfaces.

3D transformation and viewing: 3D translation - rotation and scaling - composite transformation - viewing pipeline and coordinates - parallel and perspective transformation - view volume and general (parallel and perspective) projection transformations.

UNIT V

Hidden Surface Removal: Back face detection - Z-buffer method - Painter's algorithm - scan-line algorithm - BSP-trees - Area sub-division method - Ray tracing - Animation - Morphing.

Course Outcomes

Upon completion of the course, the students will be able to:

- Differentiate various computer graphics hardware and display technologies
- Implement and develop various 2D and 3D objects
- Apply 2D and 3D transformation and viewing into the real world applications
- Project realistic view of an object
- Develop applications using Animation and Morphing

Text Books

1. D. Hearn, M. P. Baker, “Computer Graphics with OpenGL”, Fourth Edition, Pearson Education, 2013.
2. J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes, “Computer Graphics; Principles and Practice”, Second Edition in C, Addison Wesley, 1997.

Reference Books

1. D. F. Rogers, J. A. Adams, “Mathematical Elements for Computer Graphics”, Second Edition, McGraw Hill International. Edition, 1990.
2. F. S. Hill Jr., “Computer Graphics using OpenGL”, Second Edition, Pearson Education, 2003.
3. Shreiner, Woo, Neider, Davis, “OpenGL Programming Guide”, Sixth Edition, Pearson Education, 2008.
4. Peter Shirley et al., “Fundamentals of Computer Graphics”, Third Edition, A K Peters/CRC Press, 2009.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	2	0	0	0	1	0	0	0	0	0
CO2	3	0	2	0	2	2	0	0	2	0	0	2
CO3	3	0	2	0	0	2	0	0	2	0	0	2
CO4	3	0	2	0	0	0	0	0	0	1	2	0
CO5	3	0	2	1	2	2	0	0	0	0	0	2

Course Code	:	**CSPE45**
Course Title	:	**Multimedia Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand the different media and design issues in multimedia systems
- To understand communication standards for Multimedia
- Exploration of security for multimedia systems

Course Contents

UNIT I Multimedia Elements
Introduction - Definitions - Applications - Elements - Text - Image/Graphics Audio - Video - Animation.

UNIT II Audio and Speech
Data acquisition - sampling and quantization - human speech - digital model of speech production - analysis and synthesis - psychoacoustics - low bit rate speech compression - MPEG audio compression.

UNIT III Images and Video
Image acquisition and representation - bi-level image compression standards: ITU (formerly CCITT) Group III and IV standards - JPEG image compression standards - MPEG - H.264/AVC video compression standards - Transcoding.

UNIT IV Multimedia Networks
Protocol - QOS Issues - RTP - RTCP - RTSP - SIP - Media on demand - ITV - STB Broadcast Schemes for VoD Buffer Management - Multimedia over wireless networks.

UNIT V Multimedia Security and Forensics
Multimedia encryption - Digital Watermarking Security Attacks - Digital Forensics taxonomy - goals/ requirements - Forensic Data Acquisition - Forensics Analysis and Validation.

Course Outcomes

Upon completion of this course, the students will be able to:

- Design multimedia components efficiently
- Develop integrated, collaborative multimedia systems
- Understand various compression standards and techniques in multimedia
- Understand protocols for multimedia
- Develop security algorithms for the specialized applications

Text Books

1. K. Andleigh, Kiran Thakrar, “Multimedia Systems Design”, PHI, 2007.
2. ZeNian Li, S. Drew, “Fundamentals of Multimedia”, PHI, 2006.
3. Ze-Nian Li, Mark S. Drew, “Fundamentals of Multimedia”, Pearson Prentice Hall, 2004.

Reference Books

1. Jerry D. Gibson, Toby Berger, Tom Lookabaugh, Dave Lindergh, Richard L. Baker, “Digital Compression for Multimedia: Principles and Standards”, Elsevier, 2006.
2. Ralf Steinmetz, Klara, “Multimedia Computing, Communications and Applications”, Pearson Education, 2009.
3. Chun-Shien Lu, “Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property”, Springer Inc., 2007.
4. Wenjun Zeng, Heather Yu, Ching, Yung Lin, “Multimedia Security Technologies for Digital Rights Management”, Elsevier Inc., 2006.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	3	0	0	3	0	2	0	0	2	0
CO2	3	1	3	0	3	0	0	0	2	0	2	0
CO3	3	2	3	0	0	3	1	0	2	0	0	1
CO4	0	3	0	0	0	2	0	0	0	0	0	0
CO5	2	0	3	0	0	0	2	0	2	0	2	2

Course Code	:	**CSPE46**
Course Title	:	**Computing Algorithms based on Indian Knowledge Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objective:

- Understand the background of mathematical concepts developed in Indian Origin.
- Understand the application of mathematical methods in various aspects.
- Understand the benefits of these methods in view of efficient algorithm design.

Course Content:

UNIT 1:
Historical Background of main branches of mathematics - Arithmetic, Bija Grantham (Algebra), Geometry, Trigonometry, Introduction to Vedic Ganitham, Ganitha Sutras, Upa-sutras and its explanation- Dwayanka (Binary) of Maharishi Pingalacharya – 16 sutras of the binary system given in Chanda Sastram.
Prastara - Nashtam – Udistham - Samkhya - Advayoga - Eka dvi adi la ga kriya - Other minor pratyayas

UNIT 2:
Addition, subtraction - Multiplication algorithms - Ekanyunena Purvena (By one less than one before) - Ekadhikena Purvena (By one more than one Before) - Antyayordashakeapi - Nikhilam (Base 10, 100 and 1000) - Urdhvatriyakbhyam - Square - Anurupyena Method - Sankalana-vyavakalana Method - Yavadhunam Tavadhunikritya - Vargamcha Yojayet - DwandaYog - Square-root – Vilokanam - Dwandayog - Cube Anurupyena Method - Yavadhunam Tavadhunikritya Vargamcha Yojayet - Cube-root – Vilokanam.

UNIT 3:
A glimpse of Ancient Indian Ganitam - Sulaba Sutras -Boudhayana Theorem -Verification of Boudhayana Theorem- Value of $\sqrt{2}$ in Boudhayana Sulva sutra – Aryabhatta- Varahamihra – Bhrahmagupta - Bhaskaracharya -2, Vinculum Numbers (Runaank)- Method of knowing the vinculum numbers and its application.

UNIT 4:
Division – Dhawajaank – Nikhilam – Paravartya - Divisibility – Osculation, Recurring decimal, Auxiliary fraction.

UNIT 5:
Develop Algorithms based on- Bharatiya Ganitam & Chanda-Shastra-based algorithms – A case study for simple operations – A case study for complex operations - Comparison with currently implemented algorithms.

Course Outcome:

- Ability to understand the necessity and importance of mathematical concepts of Indian origin.
- Ability to develop and apply mathematical formulas based on Indian Mathematics text.
- Ability to relate the mathematical formulae of Indian origin in a real-life situation.
- Ability to develop efficient computing algorithms based on Indian Mathematics
- Ability to compare the performances of conventional computing algorithms with algorithms based on Indian mathematical concepts

Text Books:

1. Vedic Maths – Swami Bharathi Krsna Theertha, Mothilal Banarasi Das, NewDelhi.
2. Vedic Maths Nirdeshika – Part1, 2, Vidya Bharathi, Samskrit Siksha Samsthan, Kurukshetra.

Reference Books:

1. Bharath ki Pramuk Ganitacharya, Vidya Bharathi, Samskrit Siksha Samsthan, Kurukshetra.
2. Vyavaharish Kaghol Parichaya, Vidya Bharathi, Samskrit Siksha Samsthan, Kurukshetra.
3. Bharatiya Ganita Pravesha Part One, a Primer to Indian Mathematics, Samskrit Promotion Foundation, Delhi
4. Bharatiya Ganita Pravesha Part Two, Samskrit Promotion Foundation, Delhi

5. Datta B and Singh A. N., History of Hindu Mathematics, 2 vols., Lahore 1935, 1938 (repr. Asia Publishing House, Bombay 1962; repr. Bharatiya Kala Prakashan, Delhi 2004).
6. S. Balachandra Rao, Indian Mathematics and Astronomy, Bharatiya Vidya Bhavan, Bangalore, 1994.
7. C. N. Srinivasiengar, The History of Ancient Indian Mathematics, The World Press Private Limited, Calcutta, 1988.

Mapping of Course Outcomes with Programme Outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	2	1	0	3	2	3	1	3	0	1	3
CO2	3	2	1	1	3	0	0	0	0	2	1	3
CO3	3	0	0	3	3	3	0	2	0	0	1	3
CO4	1	3	3	3	2	3	2	1	2	0	1	3
CO5	1	3	3	3	3	2	3	2	0	0	0	3

V SEMESTER

Course Code	:	**CSPE51**
Course Title	:	**Augmented and Virtual Reality**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To know basic concepts of virtual reality
- To understand visual computation in computer graphics
- To understand interaction between system and computer
- To know application of VR in Digital Entertainment
- To know basic concepts of augmented reality

Course Contents

UNIT I

Introduction of Virtual Reality: Fundamental Concept and Components of Virtual Reality - Primary Features and Present Development on Virtual Reality - Multiple Models of Input and Output Interface in Virtual Reality: Input - Tracker - Sensor - Digital Glove - Movement Capture - Video-based Input - 3D Menus & 3DScanner – Output - Visual /Auditory / Haptic Devices.

UNIT II

Visual Computation in Virtual Reality: Fundamentals of Computer Graphics - Software and Hardware Technology on Stereoscopic Display - Advanced Techniques in CG: Management of Large Scale Environments & Real Time Rendering.

UNIT III

Interactive Techniques in Virtual Reality: Body Track - Hand Gesture - 3D Manus - Object Grasp.
Development Tools and Frameworks in Virtual Reality: Frameworks of Software Development Tools in VR. X3D Standard; Vega - MultiGen - Virtools.

UNIT IV

Application of VR in Digital Entertainment: VR Technology in Film & TV Production - VR Technology in Physical Exercises and Games - Demonstration of Digital Entertainment by VR.

UNIT V

Augmented and Mixed Reality: Taxonomy - technology and features of augmented reality - difference between AR and VR - Challenges with AR - AR systems and functionality - Augmented reality methods - visualization techniques for augmented reality - wireless displays in educational augmented reality applications - mobile projection interfaces - marker-less tracking for augmented reality - enhancing interactivity in AR environments - evaluating AR systems.

Course Outcomes

Upon completion of this course, the students will be able to:

- Provide opportunity to explore the research issues in Augmented Reality and Virtual Reality (AR & VR)
- Know the basic concept and framework of virtual reality
- Understand fundamentals of computer graphics
- Know the computer-human interaction
- Develop simulator for real time application using AR & VR

Text Books

1. Burdea, G. C., P. Coffet., “Virtual Reality Technology”, Second Edition, Wiley-IEEE Press, 2003/2006.
2. Alan B. Craig, “Understanding Augmented Reality, Concepts and Applications”, Morgan Kaufmann, 2013.

Reference Book

1. Alan Craig, William Sherman, Jeffrey Will, "Developing Virtual Reality Applications, Foundations of Effective Design", Morgan Kaufmann, 2009.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	2	3	0	0	3	0	1	0	1	2	2
CO2	2	0	3	0	0	0	0	0	0	0	0	0
CO3	3	0	1	0	2	2	0	0	0	1	3	0
CO4	3	2	3	2	0	3	2	0	0	0	0	2
CO5	3	2	3	2	2	3	3	1	3	1	2	3

Course Code	:	**CSPE52**
Course Title	:	**Digital Signal Processing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To introduce the various time domain to frequency domain transformations
- To learn about the IIR and FIR filter design and their representations
- To understand the issues in designing DSP systems
- To introduce multi-rate signal processing
- To know about few applications of signal processing systems

Course Contents

UNIT I Fundamentals of Signals and Systems
Basic elements of DSP - concepts of frequency in Analog and Digital Signals - sampling theorem - Discrete-time signals – systems – Analysis of discrete time LTI systems - Z transform - Convolution - Correlation.

UNIT II Frequency Transformations
Introduction to DFT - Properties of DFT - Circular Convolution - Filtering methods based on DFT - FFT Algorithms - Radix-2 - Decimation in time Algorithms - Decimation in frequency Algorithms - Use of FFT in Linear Filtering - DCT - Use and Application of DCT.

UNIT III IIR Filter Design
Structures of IIR - Analog filter design - Discrete time IIR filter from analog filter - IIR filter design by Impulse Invariance - Bilinear transformation - Approximation of derivatives (LPF - HPF - BPF - BRF) – Butterworth and Chebyshev approximation.

UNIT IV FIR Filter Design
Structures of FIR - Linear phase FIR filter - Fourier Series - Filter design using windowing techniques (Rectangular Window - Hamming Window - Hanning Window) - Binary fixed point and floating point number representations - Comparison - Quantization noise - truncation and rounding - quantization noise power - input quantization error - coefficient quantization error - limit cycle oscillations - dead band - Overflow error - signal scaling.

UNIT V Multi-Rate Signal Processing and Applications
Decimator - Interpolator - Fractional Decimation - Applications - Speech processing - Image Enhancement and Image processing system.

Course Outcomes

Upon completion of the course, students will be able to:

- Perform frequency transforms for the signals.
- Suggest appropriate IIR and FIR filter design techniques for any given application
- Predict the loss and decide on its acceptability for any application that requires digital signal processing
- Appreciate and use decimators and interpolators for any given application
- Apply DSP for Image and Speech processing systems

Text Book

1. John G. Proakis, Dimitris G. Manolakis, “Digital Signal Processing – Principles, Algorithms & Applications”, Fourth Edition, Pearson Education, Prentice Hall, 2007.

Reference Books

1. Emmanuel C. Ifeachor, Barrie W. Jervis, “Digital Signal Processing”, Second Edition, Pearson Education, Prentice Hall, 2002.
2. Sanjit K. Mitra, “Digital Signal Processing – A Computer Based Approach”, Third Edition, Tata McGraw Hill, 2007.
3. A. V. Oppenheim, R. W. Schafer, J. R. Buck, “Discrete-Time Signal Processing”, Eighth Indian Reprint, Pearson, 2004.
4. Andreas Antoniou, “Digital Signal Processing”, Tata McGraw Hill, 2006.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	1	2	0	0	0	0	0	0	0
CO2	3	3	1	1	2	0	0	0	0	0	0	1
CO3	3	3	2	1	2	0	0	0	0	0	0	1
CO4	3	3	1	1	2	0	0	0	0	0	0	0
CO5	3	3	2	1	2	1	0	0	0	0	2	1

Course Code	:	**CSPE53**
Course Title	:	**Game Theory**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To explain and predict how individuals behave in a specific strategic situation, and therefore help improve decision making
- To explain in depth the standard equilibrium concepts in Game Theory
- To illustrate the concepts, real-world examples and case studies
- To design Repeated Games with public information
- To design static and Dynamic games with incomplete information

Course Contents

UNIT I Introduction to Game Theory
Games and solutions - Game theory and mechanism design - Examples from networks - Strategic form games - Matrix and continuous games - Iterated strict dominance - Rationalizability - Nash Equilibrium - existence and uniqueness - Mixed and correlated equilibrium – Super modular games - Potential/congestion games - Existence and Properties of Nash Equilibria.

UNIT II Extensive-Form Games
Definition - Strategies and Equilibria in Extensive Form Games - Backward Induction and Subgame Perfection and its Critiques.

UNIT III Repeated Games
Infinitely/finitely repeated games - Pareto Perfection and Renegotiation - Proofness in Repeated Games - Repeated Games with incomplete Public Information - Trigger strategies - Fork Theorem with Imperfect Public Information.

UNIT IV Static Games with incomplete information
Mixed and Behavioral strategies - Bayesian Nash equilibrium - Applications in auctions - Different auction formats - Revenue and efficiency properties of different auctions - Bayesian Games and Mechanism Design Principle - Single Agent - Several Agents - Further topics in Mechanism Design.

Unit V Dynamic Games with incomplete information
Introduction - Perfect Bayesian Equilibrium in Multi-stage games - Extensive-Form and Strategic-Form Refinements - Reputation Effects - Sequential Bargaining under Incomplete Information.

Course Outcomes

Upon completion of this course, the students will be able to:
- Identify strategic situations and represent them as games
- Solve simple games using various techniques
- Recommend and prescribe which strategies to implement
- Develop Static and Dynamic Games
- Develop Repeated Games

Text Book

1. Fudenberg, Drew, Jean Tirole, “Game Theory”, Cambridge, MA: MIT Press, 1991.

Reference Books

1. Nisan, Noam, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, “Algorithmic Game Theory”, Cambridge, UK: Cambridge University Press, 2007.
2. Fudenberg, Drew, David Levine, “Theory of Learning in Games”, Cambridge, MA: MIT Press, 1998.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	2	0	3	0	0	0	3	0	0	0
CO2	2	1	0	1	0	2	0	0	0	0	2	0
CO3	0	1	0	0	0	0	0	1	0	0	2	0
CO4	1	0	1	0	2	2	0	0	2	2	2	0
CO5	1	0	1	0	2	2	0	2	0	0	3	3

Course Code	:	**CSPE54**
Course Title	:	**Real Time Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC43**
Course Type	:	**PE**

Course Objectives

- To study issues related to the design and analysis of systems with real-time constraints
- To learn the features of Real time OS
- To study the various Uniprocessor and Multiprocessor scheduling mechanisms
- To learn about various real time communication protocols
- To study the difference between traditional and real time databases

Course Contents

UNIT I Introduction to Real-time systems

Introduction to real time computing - Concepts - Example of real-time applications - Structure of a real time system - Characterization of real time systems and tasks - Hard and Soft timing constraints - Design Challenges - Performance metrics - Prediction of Execution Time: Source code analysis - Micro–architecture level analysis - Cache and pipeline issues - Programming Languages for Real-Time Systems.*

UNIT II Task Assignment and Scheduling

Real time OS - Threads and Tasks - Structure of Microkernel - Time services - Scheduling Mechanisms - Communication and Synchronization - Event Notification and Software interrupt - Uniprocessor scheduling algorithms - Task assignment - Mode changes - Fault tolerant scheduling.*

UNIT III Real–Time Communication

Network topologies and architecture issues - Protocols - Contention-based - token–based - polled bus - Fault tolerant routing.*

UNIT IV Real–Time Databases

Transaction priorities and aborts - Concurrency control issues - Scheduling algorithms – Two-phase approach to improve predictability.*

UNIT V Programming Languages and Tools

Hierarchical decomposition - Run–time error handling - Overloading - Timing specification - Recent trends and developments.*

*Programming Assignments are mandatory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Analyze scheduling problems
- Appreciate and apply Real–time programming environment tasks for solving practical problems
- Develop real time systems.
- Understand basic multi-task scheduling algorithms
- Understanding tools and utilizing them for real-time environment

Text Book

1. C. M. Krishna, Kang G. Shin, “Real–Time Systems”, International Edition, McGraw Hill Companies Inc., New York, 1997.

Reference Books

1. Jane Liu, “Real–Time Systems”, First Edition, Prentice Hall, 2002.
2. Rajib Mall, “Real–Time Systems: Theory and Practice”, First Edition, Pearson Education, 2012.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	1	0	2	2	0	0	2	0	2	0
CO2	3	1	1	1	0	2	0	0	0	0	0	2
CO3	3	2	1	0	2	0	0	0	3	0	2	0
CO4	2	3	0	0	0	0	0	2	0	0	2	0
CO5	0	2	0	0	0	0	0	0	2	0	1	0

Course Code	:	**CSPE55**
Course Title	:	**Software Testing And Automation**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- Summarize to learn the criteria for test cases
- Develop and design test cases
- Analyse test management and test automation techniques
- Assess test metrics and measurements
- Design and validate website testing

Course Contents

UNIT I Introduction

Testing as an Engineering Activity - Testing as a Process - Testing Maturity Model - Testing axioms - Basic definitions - Software Testing Principles - The Tester‘s Role in a Software Development Organization - Origins of Defects - Cost of defects - Defect Classes - The Defect Repository and Test Design - Defect Examples - Developer/Tester Support of Developing a Defect Repository.

UNIT II Test Case Design Strategies

Test case Design Strategies - Using Black Box Approach to Test Case Design - Boundary Value Analysis - Equivalence Class Partitioning - State based testing - Cause-effect graphing - Compatibility testing - user documentation testing - domain testing - Random Testing - Requirements based testing - Using White Box Approach to Test design - Test Adequacy Criteria - static testing vs. structural testing - code functional testing - Coverage and Control Flow Graphs - Covering Code Logic - Paths - code complexity testing - Additional White box testing approaches - Evaluating Test Adequacy Criteria.

UNIT III Levels of Testing

The need for Levels of Testing - Unit Test - Unit Test Planning - Designing the Unit Tests - The Test Harness - Running the Unit tests and Recording results - Integration tests - Designing Integration Tests - Integration Test Planning - Scenario testing - Defect bash elimination System Testing - Acceptance testing - Performance testing - Regression Testing - Internationalization testing - Ad-hoc testing - Alpha- Beta Tests - Testing OO systems - Usability and Accessibility testing - Configuration testing - Compatibility testing - Testing the documentation - Website testing.

UNIT IV Test Management

People and organizational issues in testing - Organization structures for testing teams - testing services - Test Planning - Test Plan Components - Test Plan Attachments - Locating Test Items - test management - test process - Reporting Test Results - Introducing the test specialist - Skills needed by a test specialist - Building a Testing Group - The Structure of Testing Group - The Technical Training Program.

UNIT V Test Automation

Software test automation - skills needed for automation - scope of automation - design and architecture for automation - requirements for a test tool - challenges in automation - Test metrics and measurements – project-progress and productivity metrics.

Course Outcomes

Upon completion of this course, the students will be able to:

- Design test cases suitable for a software development for different domains
- Identify suitable tests to be carried out
- Prepare test planning based on the document
- Document test plans and test cases designed
- Use automatic testing tools

Text Books

1. Srinivasan Desikan, Gopalaswamy Ramesh, “Software Testing – Principles and Practices”, Pearson Education, 2006.
2. Ron Patton, “Software Testing”, Second Edition, Sams Publishing, Pearson Education, 2007.

Reference Books

1. Ilene Burnstein, “Practical Software Testing”, Springer International Edition, 2003.
2. Edward Kit, “Software Testing in the Real World – Improving the Process”, Pearson Education, 1995.
3. Boris Beizer, “Software Testing Techniques”, Second Edition, Van Nostrand Reinhold, New York, 1990.
4. Aditya P. Mathur, “Foundations of Software Testing Fundamental Algorithms and Techniques”, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	2	2	0	0	0	1	0	0	1	0
CO2	3	0	0	0	2	0	0	1	0	0	1	0
CO3	0	0	0	0	3	1	0	3	0	0	3	0
CO4	2	0	0	0	0	0	0	1	0	0	2	0
CO5	1	3	0	0	0	0	0	0	0	0	2	0

Course Code	:	**CSPE56**
Course Title	:	**Cloud Computing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC43**
Course Type	:	**PE**

Course Objectives

- To provide an in-depth and comprehensive knowledge of the deployment models in Cloud Computing
- To understand the enabling technologies needed for establishing cloud environment
- To motivate students to do programming and experiment with the various cloud computing environments
- To shed light on the cloud providers and software platforms
- To introduce about different programming models in cloud computing

Course Contents

UNIT I Introduction
Evolution: Clustering - Grid computing – Virtualization – Basic concepts - Benefits and Risks - Roles and Boundaries - Characteristics - XaaS based service offerings - Basic Deployment models.

UNIT II Enabling Technologies
Networks: ISPs - Connectionless Packet Switching - Router-based Interconnectivity - Technical and Business Considerations - Data Center: Standardization and Modularity - Automation - Remote Operation - High Availability - Hardware Virtualization: Hardware Independence - Server Consolidation - Resource Replication - OS and hardware based Virtualization - Web Technology - Multitenant Technology - Service Technology.

UNIT III Computing Mechanisms
Infrastructure: Logical Network Perimeter - Virtual Server - Storage Device - Usage Monitor - Resource Replication - Specialized: Automated Scaling Listener - Load Balancer - Monitors - Failover System - Hypervisor - Resource Cluster - Multi-Device Broker - State Management Database - Management: Resource - SLA - Billing - Remote Administration - Security.

UNIT IV Cloud Providers & Software Platforms
Globally available public clouds (Microsoft Azure - Amazon Web Services - Google Cloud Platform): Overview and Comparison - Instances - Images - Networking and Security - Storage - Monitoring and Automation - Introduction to Open-source softwares: Eucalyptus - OpenNebula - OpenStack - Apache CloudStack.

UNIT V Programming Models & Advances
Introduction to MapReduce - Apache Spark - TensorFlow - Intercloud: Architecture - Resource Provisioning - Billing - Security - Mobile Cloud Computing: Resource Allocation - Security - Business Aspects - Application - Future Scope - Introduction to Edge and Fog Computing.

Course Outcomes

Upon completion of this course, the students will be able to:

- Articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing
- Identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud
- Adopt suitable computing mechanisms for establishing a cloud environment
- Provide the appropriate cloud computing solutions and recommendations according to the applications used
- Provide knowledge on recent advances and implementation of programming modes in cloud computing

Text Book

1. Kai Hwang, Geoffrey C. Fox, and Jack J. Dongarra, “Distributed and Cloud Computing from Parallel Processing to the Internet of Things”, Morgan Kaufmann, Elsevier, 2012.

Reference Books

1. Barrie Sosinsky, “Cloud Computing Bible”, John Wiley & Sons, 2010.
2. Tim Mather, Subra Kumaraswamy, Shahed Latif, “Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance”, O'Reilly, 2009.
3. James Turnbull, “The Docker Book: Containerization is the New Virtualization”, E-Book, 2015.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	3	0	0	3	0	0	0	3	2	3
CO2	0	0	0	0	3	0	0	2	0	3	3	0
CO3	3	1	3	0	0	0	0	0	2	0	0	0
CO4	3	0	3	0	0	0	2	1	0	3	2	1
CO5	0	0	3	3	2	2	3	0	0	1	1	2

Course Code	:	**CSPE57**
Course Title		**Agile Software Development**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To develop an understanding on agile software development
- To learn about the principles, planning and requirement in agile software development
- To understand the testing methodologies in agile software development
- To explore the metrics and measurement in agile software development

Course Contents

UNIT I Introduction
Agile Software Development: Basics and Fundamentals of Agile Process Methods - Values of Agile - Principles of Agile - stakeholders - Challenges Lean Approach: Waste Management - Kaizen and Kanban - add process and products add value - Roles related to the lifecycle - differences between Agile and traditional plans - differences between Agile plans at different lifecycle phases - Testing plan links between testing - roles and key techniques - principles - understand as a means of assessing the initial status of a project/ How Agile helps to build quality.

UNIT II Principles
Agile and Scrum Principles: Agile Manifesto - Twelve Practices of XP - Scrum Practices - Applying Scrum - Need of scrum - working of scrum - advanced Scrum Applications - Scrum and the Organization - scrum values.

UNIT III Planning and Product Management
Agile Product Management: Communication - Planning - Estimation - Managing the Agile approach - Monitoring progress - Targeting and motivating the team - Managing business involvement - Escalating issue - Quality - Risk - Metrics and Measurements.

UNIT IV Requirements and Testing
Agile Requirements: User Stories - Backlog Management - Agile Architecture: Feature Driven Development - Agile Risk Management: Risk and Quality Assurance - Agile Tools - Agile Testing: Agile Testing Techniques - Test-Driven Development - User Acceptance Test - Agile Review: Agile Metrics and Measurements - The Agile approach to estimating and project variables.

UNIT V Measurement
Agile Measurement - Agile Control-control parameters - Agile approach to Risk - The Agile approach to Configuration Management - The Atern Principles - Atern Philosophy - Rationale for using Atern - Refactoring - Continuous integration - Automated Build Tools - Scaling Agile for large projects: Scrum of Scrums - Team collaborations - Scrum - Estimate a Scrum Project - Track Scrum Projects - Communication in Scrum Projects - Best Practices to Manage Scrum.

Course Outcomes

Upon completion of this course, the students will be able to:

- Distinguish between agile software development and traditional software development
- Suggest agile software development approaches for any real-time problem
- Design and provide measurement, metrics necessary for problems involving agile software development
- Integrate best practices of traditional and agile software development and use in real-time problem solving
- Estimate risk of scrum projects

Text Books

1. Robert C. Martin, “Agile Software Development, Principles, Patterns, and Practices”, First Edition, Pearson Education India, 2002.
2. “Succeeding with Agile: Software Development Using Scrum”, Pearson, 2010.

Reference Book

1. Robert C Martin, Micah Martin, “Agile Principles, Patterns and Practices in C#”, Pearson Education, 2007.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	2	0	0	0	0	0	2	0	3	3	0
CO2	0	0	2	0	0	2	0	0	0	2	3	0
CO3	0	0	0	0	3	0	0	1	3	3	3	0
CO4	2	0	2	0	0	0	0	2	0	0	1	0
CO5	0	2	3	0	3	0	0	2	3	0	2	0

Course Code	:	**CSPE58**
Course Title	:	**Data Interpretation and Analysis**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To provide fundamental concepts in statistics and probability
- To study random variables and various distributions
- To understand parameter estimation and hypothesis testing
- To learn regression and its different types of analysis models

Course Content:

Unit -1

Descriptive statistics: Population, sample, parameter, sampling, concept of frequency, frequency tables and graphs. Summarizing numerical data: mean, median, mode, percentile, variance, standard deviation. Chebyshev inequality with proof, correlation coefficient, correlation and causation, proof of one-sided Chebyshev's inequality. **Probability**- Sample space, event, De Morgan's laws, Boole's and Bonferroni's inequalities, conditional probability, Bayes rule, false positive paradox, Birthday paradox in discrete probability

Unit-2

Random Variables: Discrete and continuous random variables: mean, median, moments, variance. Probability mass function (pmf), cumulative distribution function (cdf) and probability density function (pdf). Discrete RVs: Bernoulli, Binomial, Geometric, Indicator. Continuous RVs, Joint distributions and conditioning. Law of the Unconscious Statistician (LOTUS). Markov's and Chebyshev's inequality. Weak law of large numbers, Gambler's fallacy, Moment generating function.

Unit-3

Special Random Variables: Bernoulli PMF, Binomial PMF, Gaussian PDF: central limit theorem, Expression for CDF and its relation to the error function, de Moivre-Laplace theorem. Derivation of PDF of mean of different random variables; Bessel's correction for standard deviation; PDF of sample mean and sample variance of a Gaussian. Chi square distribution, Uniform distribution, Poisson distribution; Exponential distribution. Multinomial PMF - generalization of the binomial, mean vector and covariance matrix for a multinomial random variable, MGF for multinomial.

Unit-4

Parameter Estimation: Concept of parameter estimation, Maximum likelihood estimation (MLE), MLE for parameters of Bernoulli, Poisson, Gaussian and uniform distributions. Least squares line fitting as an MLE problem. Concept of estimator bias, mean squared error, variance; Concept of two-sided confidence interval and one-sided confidence interval. Nonparametric density estimation. Concept of histogram as a probability density estimator; Bias, variance and MSE for a histogram estimator for a smooth density. Hypergeometric distribution: genesis, mean, variance. Concept of kernel density estimator; Bias, variance and MSE for a kernel density estimator for a smooth density.

Unit- 5

Hypothesis Testing: Significance levels, tests concerning the mean of normal population, Bernoulli population, and Poisson distribution, testing the equality of means of two normal populations, hypothesis tests for variance of a normal population. **Regression:** Least square estimators of regression parameters; Distribution of the estimators,

statistical inference about regression parameters; Coefficient of determination and the sample correlation coefficient. Analysis of residuals, weighted least squares, polynomial regression, multiple linear regression, logistic regression models.

Course outcomes:

- Comprehend the essential concepts in probability theory
- Use mathematical tools for analyzing probabilities
- Applying the appropriate concepts of probability and statistics to solve real-world problems
- Understand various distributions and hypothesis tesing
- Understand various regression models and their applications.

Text book:

1. Sheldon M Ross, Introduction to Probability and Statistics for Engineers and Scientists, 5th edition, Elsevier Inc 2014

Reference Books:

1. M. H. DeGroot and M. J. Schervish, Probability and Statistics, 4th edition, Addison-Wesley, 2012
2. Jay L. Devore , Probability and Statistics for Engineering and the Sciences. 9th edition, 2012
3. Larry Wasserman , All of Statistics: A Concise Course in Statistical Inference, 1st edition, Springer publication, 2004

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	2	3	0	0	0	0	0	0	0	0
CO2	3	2	2	3	0	1	0	0	0	0	1	1
CO3	1	3	2	2	0	2	0	0	0	0	1	1
CO4	1	1	1	1	0	0	0	0	0	0	0	0
CO5	2	2	3	1	0	1	0	0	0	0	1	1

VI SEMESTER

Course Code	:	**CSPE61**
Course Title	:	**Web Technology and its Applications**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC42**
Course Type	:	**PE**

Course Objectives

- To understand the basics of Web Designing using HTML, DHTML, and CSS
- To learn the basics about Client side scripts and Server side scripts

Course Contents

UNIT I
HTML - Introduction - HTML Formatting - Hyper-Links - Lists - Tables - Images - Forms - Frames - Cascading Style sheets - Types - XML - Document type definition - XML Schemas - Document Object model.*

UNIT II
Introduction to Client Side scripting - JavaScript - Control statements - Functions - Arrays - Objects - Events - Dynamic HTML with Java Script - AJAX: Ajax Client Server Architecture - XML Http Request Object - Call Back Methods.*

UNIT III
NodeJS and Express - Introduction to AngularJS and Fundamentals of ReactJS - Web servers – IIS (XAMPP - LAMPP) and Tomcat Servers - Server Side Scripting - Java Servlets - Java Server Pages - Java Server Faces - JSF Components - Session Tracking - Cookies.*

UNIT IV
PHP - Basic Syntax - Defining variable and constant - PHP Data types - Operator and Expression - Operator Precedence - Decisions and Loop - Functions & Recursion - String Processing and Regular Expressions - Form Processing - Working with file and Directories* - Cookies.

UNIT V
Database Connectivity with MySQL - Servlets - JSP - PHP - MongoDB - NOSQL Database* - Fundamentals of JQuery and Bootstrap.

*Programming assignments are mandatory.

Course Outcomes

Upon completion of the course, the students will be able to:

- Understand and interpret standard web technologies
- Build real world applications using client side and server side scripting languages
- Design and develop applications using web technologies
- Suggest appropriate web technologies for any application
- Handling web application data with databases

Text Books

1. Paul J. Deitel, Harvey M. Deitel, Abbey Deitel, “Internet & World Wide Web How to Program”, Fifth Edition, Deitel Series, 2012.
2. Jason Gilmore, “Beginning PHP and MySQL from Novice to Professional”, Fourth Edition, Apress Publications, 2010.

3. Brown, Ethan, “Web Development with Node and Express: Leveraging the JavaScript Stack”, O'Reilly Media, 2019.
4. Anthony, Accomazzo, Murray Nathaniel, Lerner Ari, “Fullstack React: The Complete Guide to React JS and Friends”, Fullstack.io, 2017.
5. Kozlowski, Pawel, “Mastering Web Application Development with Angular JS”, Packt Publishing Ltd., 2013.

Reference Books

1. Robert W. Sebesta, “Programming with World Wide Web”, Fourth Edition, Pearson, 2008.
2. David William Barron, “The World of Scripting Languages”, Wiley Publications, 2000.
3. Dayley B., “Node.js, MongoDB, and AngularJS Web Development”, Addison-Wesley Professional, 2014.
4. Vainikka J., “Full-Stack Web Development using Django REST Framework and React”, 2018.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	1	0	2	0	0	0	2	0	0	2	0
CO2	3	0	3	0	0	2	0	0	0	3	3	0
CO3	0	0	3	0	3	3	0	0	0	3	3	0
CO4	0	2	3	0	0	0	0	0	0	0	3	0
CO5	3	0	0	1	0	0	0	2	0	0	0	0

Course Code	:	**CSPE62**
Course Title	:	**Advanced Database Management Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC52**
Course Type	:	**PE**

Course Objectives

- To understand the different database models and language queries to access databases
- To understand the normalization forms in building an effective database tables
- To protect the data and the database from unauthorized access and manipulation

Course Contents

UNIT I Relational Model Issues
ER Model - Normalization - Query Processing - Query Optimization - Transaction Processing - Concurrency Control - Recovery - Database Tuning.

UNIT II Distributed Databases
Parallel Databases - Inter and Intra Query Parallelism - Distributed Database Features - Distributed Database Architecture - Fragmentation - Distributed Query Processing - Distributed Transactions Processing - Concurrency Control - Recovery - Commit Protocols.*

UNIT III Object Oriented Databases
Introduction to Object Oriented Data Bases - Approaches Modelling and Design - Persistence - Query Languages - Transaction - Concurrency - Multi Version Locks - Recovery - POSTGRES - JASMINE - GEMSTONE - ODMG Model.*

UNIT IV Emerging Systems with Current Issues
Enhanced Data Models - Client/Server Model - Data Warehousing and Data Mining - Web Databases - Mobile Databases - XML and Web Databases - Rules - Knowledge Bases - Active and Deductive Databases - Multimedia Databases - Multimedia Data Structures - Multimedia Query languages - Spatial Databases.*

UNIT V Advanced Databases
Introduction to NoSQL - Aggregate Data Models - Hbase: Data Model and Implementations - Hbase Clients – Examples – Cassandra: Data Model – Examples – Cassandra Clients - Hadoop Integration - Traditional Indexing Methods (Secondary Keys - Spatial Access Methods) - Text Retrieval - Multimedia Indexing - 1D Time Series - 2D Color images – Sub pattern Matching – Open Issues - Uncertainties.

*Programming assignments are mandatory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Comprehend the complex query processing techniques
- Design distributed and object oriented databases
- Comprehend emerging and advanced databases
- Design and implement multimedia databases and writing query structure
- Develop skill set in file organization, Query Optimization, Transaction management, and database administration techniques

Text Books

1. Thomas Connolly, Carlolyn Begg, “Database Systems: A Practical Approach to Design, Implementation, and Management”, Fifth Edition, Addison-Wesley, 2009.
2. R. Elmasri, S. B. Navathe, “Fundamentals of Database Systems”, Fifth Edition, Pearson/Addison Wesley, 2006.
3. P. J. Sadalage, M. Fowler, “NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence”, Addison-Wesley Professional, 2012.
4. Lars George, “HBase: The Definitive guide”, O'Reilly Media Inc., 2011.
5. Eben Hewitt, “Cassandra: The definitive Guide”, O'Reilly Media Inc., 2010.
6. Carlo Zaniolo, Stefano Ceri, “Advanced Database Systems”, Morgan Kauffmann Publishers.

Reference Books

1. Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Fifth Edition, Tata McGraw Hill, 2006.
2. C. J. Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	0	0	0	0	0	0	0	1	2	0
CO2	2	2	1	0	1	2	0	0	1	2	2	2
CO3	1	0	1	0	1	0	0	0	0	2	0	3
CO4	0	3	2	0	3	0	0	0	2	2	2	0
CO5	2	0	2	0	0	0	0	2	2	2	2	0

Course Code	:	**CSPE63**
Course Title	:	**Artificial Intelligence and its Applications**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC54**
Course Type	:	**PE**

Course Objectives

- To learn about agents and knowledge representation
- To understand the various logical reasoning systems
- To familiarize in planning activities for agents
- To explore the uncertainty involves in reasoning systems
- To get introduced to communicating agents and robotics

Course Contents

Unit I
Fundamentals of AI - Search techniques review - knowledge representation review - logical agents - simple reflex agent - Building knowledge base - Knowledge Engineering - General ontology - Representing categories - measures - composite objects - Resolution - complete inference procedure.*

Unit II
Logical Reasoning systems - indexing - retrieval and unification - theorem provers - forward-chaining production systems - frame systems and semantic networks - Certainty factors - Bayesian Theory - Bayesian Network - Dempster - Shafer theory.*

Unit III
Planning - simple planning agent - representation for planning - partial-order planning - Planning with partial instantiated operators - Knowledge engineering for planning - practical planners - hierarchical decomposition - expressive operator decomposition - Conditional planning - simple replanning agent - Fully integrated planning and execution.*

Unit IV
Uncertainty - probabilistic reasoning systems - representing knowledge in uncertain domain - semantics of belief networks - inference in belief networks - inference in multiple connected belief networks - Knowledge engineering for uncertain reasoning.*

Unit V
Agents that communicate - Types of communicating agents - Introduction to Robotics - Architectures - configuration spaces - navigation and motion planning.*

*Programming assignments are mandatory.

Course Outcomes

Upon completion of the course, the students will be able to:
- Design agents for any given problem
- Suggest appropriate knowledge representation technique for any problem
- Develop algorithms involving planning agents
- Appreciate the uncertainty in designing AI systems and propose algorithms for the same
- Propose new architectures for communicating agents

Text Books

1. Stuart Russel, Peter Norvig, “AI – A Modern Approach”, Second Edition, Pearson Education, 2007.
2. Kevin Night, Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, McGraw Hill, 2008.

Reference Books

1. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education, 2007.
2. Peter Jackson, “Introduction to Expert Systems”, Third Edition, Pearson Education, 2007.
3. Stuart Russel, Peter Norvig, “AI – A Modern Approach”, Second Edition, Pearson Education, 2007.
4. G. Luger, W. A. Sttubblefield, “Artificial Intelligence”, Third Edition, Addison-Wesley Longman, 1998.
5. N. J. Nilson, “Principles of Artificial Intelligence”, Narosa Publishing House, 1980.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	2	0	0	1	0	0	0	0	0	0
CO2	0	3	2	0	1	0	0	0	1	0	2	0
CO3	0	2	0	3	2	0	0	0	1	0	2	0
CO4	3	2	2	0	1	0	0	0	1	0	0	2
CO5	0	0	2	0	0	3	0	0	0	0	1	0

Course Code	:	**CSPE64**
Course Title	:	**Data Analytics**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course code)	:	**CSPC52**
Course Type	:	**PE**

Course Objectives

- To understand the basic principles of Data Analytics
- To learn the various Data Analytic methods
- To understand the various clustering algorithms and its application on data
- To work with stream data model and computing

Course Contents

UNIT I
Introduction to Data Analytics - Types of Data Analytics - Predictive Analytics - Simple linear regression - Multiple linear regression - Auto regression - Moving Average - Autoregressive Integrated Moving Average - Data Pre-processing - Data Cleaning - Data Integration and Transformation - Data Reduction - Descriptive data analytics - measures of central tendency - measures of location of dispersions.

UNIT II
Association Rule Mining: Efficient and Scalable Frequent Item set Mining Methods - Mining Various Kinds of Association Rules - Association Mining to Correlation Analysis - Constraint Based Association Mining - Cluster Analysis: Types of Data in Cluster Analysis - A Categorization of Major Clustering Methods - Partitioning Methods - Hierarchical methods.

UNIT III
Introduction to Streams Concepts - Stream data model and architecture - Stream Computing - Sampling data in a stream - Filtering streams - Counting distinct elements in a stream - Estimating moments - Counting oneness in a window - Decaying window - Real Time Analytics Platform (RTAP) applications - case studies - real time sentiment analysis - stock market predictions.

UNIT IV
Using Graph Analytics for Big Data: Graph Analytics - The Graph Model - Representation as Triples - Graphs and Network Organization - Choosing Graph Analytics - Graph Analytics Use Cases - Graph Analytics Algorithms and Solution Approaches - Technical Complexity of Analyzing Graphs - Features of a Graph Analytics Platform - Considerations: Dedicated Appliances for Graph - Graph QL

UNIT V
NoSQL Databases - Schema-less Models - Increasing Flexibility for Data Manipulation - Key Value Stores - Document Stores - Tabular Stores - Object Data Stores - Graph Databases Hive-Sharding-Hbase - Analyzing big data with twitter - Big data for E-Commerce - Big data for blogs - Review of Basic Data Analytic Methods using R.

Course Outcomes

Upon completion of this course, the students will be able to:

- Evaluate the use of data from acquisition through cleaning, warehousing, analytics, and visualization to the ultimate business decision
- Mine data and carry out predictive modeling and analytics to support business decision-making
- Suggest prescriptive modeling techniques for real-world problems

- Execute real-time analytical methods on streaming datasets to react quickly to customer needs
- Apply graph analytics on data

Text Books

1. Jiawei Han, MichelineKamber, Jian Pei, "Data Mining Concepts and Techniques", Third Edition, Elsevier, 2011.
2. A. Rajaraman, J. Ullman, "Mining Massive Data Sets", Cambridge University Press, 2012.
3. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, No SQL, and Graph", 2013.

Reference Books

1. Ronald E. Walpole, Raymond H. Myers, Sharon L. Myers, Keying Ye, "Probability & Statistics for Engineers & Scientists", Ninth Edition, Prentice Hall Inc.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, "The Elements of Statistical Learning, Data Mining, Inference, and Prediction", Second Edition, Springer, 2014.
3. G James, D. Witten, T Hastie, R. Tibshirani, "An Introduction to Statistical Learning: With Applications in R", Springer, 2013.
4. Mohammed J. Zaki, Wagner Meira, "Data Mining and Analysis", Cambridge, 2012.
5. E. Alpaydin, "Introduction to Machine Learning", MIT Press, 2014.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	1	0	0	0	0	2	0	0	2	0
CO2	3	2	3	0	0	3	0	1	0	0	3	0
CO3	3	2	2	0	0	3	0	0	0	0	3	0
CO4	3	2	3	0	1	2	0	0	1	0	2	1
CO5	3	2	3	0	0	3	0	1	3	0	3	0

Course Code	:	**CSPE65**
Course Title	:	**Machine Learning Techniques and Practices**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC54**
Course Type	:	**PE**

Course Objectives

- To understand the principles and concepts of machine learning
- To learn the clustering techniques and their utilization in machine learning
- To study the neural network systems for machine learning
- To understand reinforcement and deep learning
- To learn methodology and tools to apply machine learning algorithms to real data and evaluate their performance

Course Contents

UNIT I

Introduction and mathematical preliminaries - Vectors - Inner product - Outer product - Inverse of a matrix - Eigen analysis - Singular value decomposition - Probability distributions - Conditional probability distribution and Joint probability distribution - Bayes theorem - Types of Machine Learning - Supervised Learning - Classification models - Naïve Bayes Classifier - Decision trees - Entropy computation using GINI - Information Gain - Support Vector Machines - non-linear kernels - KNN model - MLP - CART - Ensemble Methods: Bagging - Boosting - Gradient boosting.

UNIT II

Linear models for regression - Maximum Likelihood Estimation (MLS) - least squares - regularized least squares - The Bias-Variance Decomposition - Bayesian Linear Regression - Linear models for classification - Discriminant functions - Fisher's linear discriminant - Probabilistic generative models - Probabilistic discriminative models - Bayesian logistic regression - Bayesian learning - maximum aposterior (MAP) estimation.

UNIT III

Clustering - K-Means clustering - Hierarchical Clustering - Mixture of Gaussians - Expectation maximization for mixture models (EM) - Dimensionality Reduction - Principal Component Analysis (PCA) - Linear Discriminant Analysis (LDA).

UNIT IV

Graphical models - Markov random fields - Hidden Markov Models - Representation - learning - Decoding - Inference in graphical models - Monte Carlo models - Sampling.

UNIT V

Reinforcement Learning - Model based - Model Free - Q learning - Introduction to Deep learning.

Course Outcomes

Upon completion of this course, the students will be able to:

- Appreciate the underlying mathematical relationships within and across machine learning algorithms and the paradigms of supervised and un-supervised learning
- Have an understanding of the strengths and weaknesses of machine learning algorithms
- Appreciate machine learning challenges and suggest solutions for the same
- Design and implement various machine learning algorithms in a range of real-world applications
- Suggest supervised / unsupervised machine learning approaches for any application

Text Books

1. Tom Mitchell, "Machine Learning", McGraw Hill, 1997.
2. E. Alpaydin, "Introduction to Machine Learning", Second Edition, Prentice-Hall of India, 2010.

Reference Books

1. Simon Haykin, “Neural Networks and Learning Machines”, Pearson, 2008.
2. Shai Shalev-Shwartz, Shai Ben-David, “Understanding Machine Learning from Theory to Algorithms”, Cambridge University Press, 2014.
3. R.O. Duda, P.E. Hart, D.G. Stork, “Pattern Classification”, Second Edition, Wiley-Inderscience, 2000.
4. T. Hastie, R. Tibshirani, J. Friedman, “The Elements of Statistical Learning”, Springer, 2011.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	0	0	0	0	0	0	0	0	0	0
CO2	0	0	3	0	0	2	0	0	0	0	0	0
CO3	0	3	0	0	0	1	0	0	0	0	0	0
CO4	0	0	3	1	0	2	0	0	0	0	0	0
CO5	0	0	3	3	0	1	0	0	0	0	0	0

Course Code	:	**CSPE66**
Course Title	:	**Mobile Computing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC53**
Course Type	:	**PE**

Course Objectives

- To understand the fundamentals of mobile communication
- To understand the architecture of various Wireless Communication Networks
- To understand the significance of different layers in mobile system

Course Contents

UNIT I
Introduction to Wireless Networks - Applications - History - Simplified Reference Model - Wireless transmission - Frequencies - Signals - Antennas - Signal propagation - Multiplexing - Modulation - Spread spectrum - Cellular Systems: Frequency Management and Channel Assignment - types of hand-off and their characteristics.*

UNIT II
MAC - Motivation - SDMA - FDMA - TDMA - CDMA - Telecommunication Systems - GSM: Architecture- Location tracking and call setup - Mobility management - Handover - Security - GSM - SMS - International roaming for GSM - call recording functions - subscriber and service data management - DECT - TETRA - UMTS - IMT-2000.*

UNIT III
Wireless LAN - Infrared vs. Radio transmission - Infrastructure - Adhoc Network - IEEE 802.11WLAN Standards - Architecture - Services - HIPERLAN - Bluetooth Architecture & protocols.*

UNIT IV
Mobile Network Layer - Mobile IP - Dynamic Host Configuration Protocol - Mobile Transport Layer - Traditional TCP - Indirect TCP - Snooping TCP - Mobile TCP - Fast retransmit/Fast recovery - Transmission/Time-out freezing - Selective retransmission - Transaction Oriented TCP.*

UNIT V
WAP Model - Mobile Location based services - WAP Gateway - WAP protocols - WAP user agent profile caching model - wireless bearers for WAP - WML - WML Scripts – WTA – iMode - SyncML.*

*Programming assignments are mandatory.

Course Outcomes

Upon completion of the course, the students will be able to:

- Develop a strong grounding in the fundamentals of mobile Networks
- Define mobile technologies in terms of hardware, software, and communications
- Apply knowledge in MAC, Network, and Transport Layer protocols of Wireless Network
- Demonstrate the Adhoc networks concepts and its routing protocols
- Comprehend, design, and develop a lightweight network stack

Text Books

1. Jochen Schiller, “Mobile Communication”, Second Edition, Pearson Education, 2008.
2. Theodore, S. Rappaport, “Wireless Communications, Principles, Practice”, Second Edition, PHI, 2002.

Reference Books

1. William Stallings, “Wireless Communications and Networks”, Second Edition, Pearson Education, 2004.
2. C. Siva Ram Murthy, B. S. Manoj, “Adhoc Wireless Networks: Architectures and Protocols”, Second Edition, Pearson Education, 2008.
3. Vijay. K. Garg, “Wireless Communication and Networking”, Morgan Kaufmann Publishers, 2007.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1		1							1	
CO2	1	2			1							
CO3	2	2				2			2		2	2
CO4	3	2										
CO5	3	2				2			2		2	

Course Code	:	**CSPE67**
Course Title	:	**Internetworking Protocols**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC53**
Course Type	:	**PC**

Course Objectives

- To provide insight about networks, topologies, and the key concepts
- To gain comprehensive knowledge about the layered communication architectures (OSI and TCP/IP) and its functionalities
- To understand the principles, key protocols, design issues, and significance of each layers in ISO and TCP/IP
- To know the implementation of various layers

Course Contents

UNIT I Network Topology
Review of Reference Models - Topology and switching - IEEE Standard 802 from Ethernet - Token Bus - Token Ring and Wireless LAN - Connecting Devices.

UNIT II Introduction to IPV4
IPv4 headers - IP forwarding - Host Processing of IP datagrams - DHCP and Auto-configuration - Firewalls and NAT - ICMPv4 - IP Fragmentation - DNS - Broadcasting and Local Multicasting - IGMP - Routing Protocols.

UNIT III IPV6
IPv6 Transition issues - Protocol basics - Addressing - Options and Extension headers - ICMPv6 - Neighbor Discovery - Routing - Autoconfiguration - Multicast Listener Discovery (MLD) - IPv6 and DNS.

UNIT IV Transmission Control Protocol
Transmission Control Protocol (TCP) - TCP Connection Management - TCP Data Flow and Window Management - Stream Control Transmission Protocol (SCTP) - Services - SCTP Association management - SCTP flow and error control.

UNIT V Overview of Mobile IP
Need for Mobile IP - Overview of Mobile IP - Details of Mobile IP - Tunneling - Mobility for IPv6 - Applications of Mobile IP – Security primer - Campus Mobility - Internet wide mobility - A service provider perspective.

Course Outcomes
Upon completion of this course, the students will be able to:

- Gain insight about basic network theory and layered communication architectures
    - Configure and troubleshoot basic IPv4 network settings, such as IP addresses, subnet masks, default gateways, and DNS servers
    - Familiarize about the basics of IP addressing, subnetting, and routing in IPv6 networks
    - Code and implement different types of transport layer protocols such as UDP, TCP and SCTP
- Design and develop Mobile IP

Text Books

1. W. Richard Stevens, G. Gabrani, “TCP/IP Illustrated: The Protocols”, Pearson, 2011.
2. Peter Loshin, Morgan Kaufmann, “IPv6: Theory, Protocol, and Practice”, Second Edition, 2003.

Reference Books

1. James Solomon, "Mobile IP: The Internet Unplugged", First Edition, Pearson Education, 2008.
2. Kevin R. Fall, W. Richard Stevens, "TCP/IP Illustrated, Vol. 1 - The Protocols", Second Edition, Addison-Wesley, 2012.
3. Silvia Hagen, "IPv6 Essentials", Second Edition, O'Reilly Media, 2006.
4. Charles E. Perkins, "Mobile IP: Design Principles and Practices", First Edition, Pearson Education, 2008.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	1	0	0	0	2	0	0	0	3	0
CO2	1	2	0	0	0	0	2	0	0	0	3	0
CO3	1	2	2	0	0	2	2	0	0	0	3	0
CO4	0	1	1	0	2	0	0	0	0	3	0	0
CO5	1	1	0	2	0	2	0	0	0	0	0	3

Course Code	:	**CSPE68**
Course Title	:	**Design and Analysis of Parallel Algorithms**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC42**
Course Type	:	**PE**

Course Objectives

- To understand different parallel architectures and models of computation
- To introduce the various classes of parallel algorithms
- To study parallel algorithms for basic problems
- To study graph Algorithms

Course Contents

UNIT I Introduction to Parallel Computers
Need for parallel processing - SM-SIMD algorithms - Shared memory SIMD - Tree and mesh interconnection computers - Classifying MIMD Algorithms - parallel computational models such as PRAM - LMCC - Hypercube - Cube Connected Cycle - Butterfly - Perfect Shuffle Computers - Tree model - Pyramid model - Fully Connected model - PRAM - CREW - EREW models - simulation of one model from another one.

UNIT II Matrix Operations and Performance Measures
Matrix operations - Mesh transpose - Shuffle transpose - EREW transpose - Mesh multiplication - Cube multiplication - Matrix by vector multiplication - Tree multiplication - Performance Measures of Parallel Algorithms - speed-up and efficiency of PA - Cost optimality - An example of illustrate Cost-optimal algorithms such as summation - Min/Max on various models.

UNIT III Selection and Sorting
Sequential algorithm - Parallel Sorting Networks - Algorithm for parallel selection - Sorting on a linear array - broadcasting a datum - Computing all sums - Sorting on a mesh - Sorting on EREW SIMD computer - enumeration sort - parallel quick sort - hyper quick sort - Sorting on other networks - Parallel Merging Algorithms on CREW/EREW/MCC/ - Parallel Sorting Networks on CREW/EREW/MCC.

UNIT IV Searching and Numerical Problems
Parallel Searching Algorithm - Kth element - Kth element in X+Y on PRAM - Parallel Matrix Transportation and Multiplication Algorithm on PRAM - MCC - Vector-Matrix Multiplication - Solution of Linear Equation - Root finding - SIMD algorithm - Roots of nonlinear equations - MIMD algorithm - Partial differential equations - Computing Eigen values - Monte Carlo methods - parallel random number generators - random number distributions.

UNIT V Graph Problems
Definitions - Graph coloring - Computing the connectivity matrix - Finding connected components - search and traversal - Minimal alpha-beta tree - Minimum Cost Spanning Tree - Addition tree - Multiplication tree - Combinatorial Algorithms - Permutation - Combinations - Derrangements.

Course Outcomes

Upon completion of this course, the students will be able to:

- Develop parallel algorithms for standard problems and applications
- Explain and derive the complexity of algorithms for basic and collective communication operations
- Apply different methods and performance measures to analyze algorithms with respect to cost and scalability
- Describe the basic methods of problem and data partitioning for efficient memory utilization and minimization of communication costs in parallel computers
- Perform design and analysis of parallel algorithms in real time applications

Text Books

1. M. J. Quinn, “Designing Efficient Algorithms for Parallel Computer”, McGraw Hill.
2. S. G. Akl, “Design and Analysis of Parallel Algorithms”, PHI, 1989.

Reference Books

1. J. Jaja, “An Introduction to Parallel Algorithms”, Addison Wesley, 1992.
2. F. T. Leighton, “Introduction to Parallel Algorithms and Architectures: Arrays, Trees”.
3. “Hypercubes”, Morgan Kaufmann Publishers, San Mateo, California, 1992.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	3	0	0	2	0	0	0	0	0	0
CO2	3	2	2	0	3	0	0	0	2	0	2	0
CO3	3	3	0	3	0	3	0	0	2	0	2	0
CO4	3	3	0	0	3	2	0	0	3	0	2	0
CO5	2	0	3	0	0	2	0	0	0	0	0	2

Course Code	:	**CSPE69**
Course Title	:	**Brain Computer Interface and its Applications**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Courses Objectives:

The student should be made to:

• Understand the basic concepts of brain computer interface
• Study the various signal acquisition methods
• Learn about the signal processing methods used in BCI
• Understand the various machine learning methods of BCI.
• Learn the various applications of BCI

Course Content:

UNIT I INTRODUCTION TO BCI

Introduction - Brain structure and function, brain-computer interface Types - Synchronous and Asynchronous - Invasive BCI - Partially Invasive BCI – Non-Invasive BCI, Structure of BCI System, BCI Monitoring Hardware, BCI Techniques - EEG, ECoG, MEG, fMRI, Ethics of Brain-Computer Interfacing.

UNIT II SIGNAL ACQUISITION TECHNIQUES

Brain activation patterns - brainwaves, oscillatory potential and event related potential, error related potential, slow cortical potentials, movement related potentials, stimulus related potentials - visual evoked potentials – P300 and auditory evoked potentials, potentials related to cognitive tasks.

UNIT III FEATURE EXTRACTION AND ANALYSIS

Signal pre-processing – filtering techniques, artefacts reduction, frequency domain analysis, wavelet analysis, time domain analysis, spatial filtering - Principal Component Analysis (PCA), Independent Component Analysis (ICA), topographical maps.

UNIT IV MACHINE LEARNING TECHNIQUES FOR BCI

Classification techniques – binary classification, ensemble classification, multiclass classification, Classifiers - support vector machine, neural networks, Regression – trees, deep learning techniques for BCI, evaluation metrics for BCI applications.

UNIT V APPLICATIONS OF BCI

Case Studies: Medical applications - sensory restoration, motor restoration, cognitive restoration, rehabilitation, restoring communication with menus, cursors, and spellers, brain-controlled wheelchairs; Non-medical applications- web browsing and navigating virtual worlds, high throughput image search lie detection and applications in law, imagined thoughts, emotion recognition.

Course Outcomes:

At the end of the course, the student should be able to:

• Comprehend and appreciate the significance and role of this course in the present contemporary world.
• Evaluate the concept of BCI.
• Assign functions appropriately to the human and to the machine.
• Select appropriate feature extraction and analysis methods
• Use machine learning algorithms for translation.

Text Books:

1. Rajesh. P. N. Rao, “Brain-Computer Interfacing: An Introduction”, Cambridge University Press, First edition, 2013.
2. Jonathan Wolpaw, Elizabeth Winter Wolpaw, “Brain Computer Interfaces: Principles and practice”, Oxford University Press, USA, Edition 1, January 2012.

Reference Books:

1. Ella Hassianien, A &Azar.A.T (Editors), “Brain-Computer Interfaces Current Trends and Applications”, Springer, 2015.
2. Bernhard Graimann, Brendan Allison, GertPfurtscheller, "Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction", Springer, 2010
3. Saeid Sanei, Jonathon A. Chambers, “EEG Signal Processing and Machine Learning”, Wiley International, 2nd Edition, 2021
4. Arnon Kohen, “Biomedical Signal Processing”, Vol I and II, CRC Press Inc, Boca Rato, Florida, 2021.
5. Bishop C.M., “Neural networks for Pattern Recognition”, Oxford, Clarendon Press, 1995.
6. Andrew Webb, “Statistical Pattern Recognition”, Wiley International, Second Edition, 2002.
7. Mike X Cohen, “Analyzing Neural Time Series Data: Theory and Practice”, The MIT Press, First edition, 2014.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	3	0	0	2	2	0	0	1	0	0	2
CO2	1	2	0	3	2	3	0	0	0	0	2	0
CO3	2	2	2	3	2	3	3	1	1	0	0	1
CO4	1	1	2	3	1	1	2	0	0	0	0	0
CO5	1	3	1	1	0	2	3	0	0	0	0	0

VII SEMESTER

Course Code	:	**CSPE71**
Course Title	:	**Advanced Cryptography**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC63**
Course Type	:	**PE**

Course Objectives

- To study the concepts of applied cryptography
- To understand the application of cryptographic techniques in real world applications
- To comprehend the notion of provable security and its implication with improved security guarantees
- To introduce the concept of block chain technology

Course Contents

UNIT I
Review of number theory - group - ring and finite fields - quadratic residues - Legendre symbol - Jacobi symbol - Probability - Probability - Discrete random variable - Continuous random variable - Markov's inequality - Chebyshev's inequality - normal distribution - the geometric and binomial distributions.*

UNIT II
Symmetric cryptosystem - Stream cipher - Cryptanalysis: Cube attack - Formal Notions of Attacks: Attacks under Message Indistinguishability: Chosen Plaintext Attack (IND-CPA) - Chosen Cipher text Attacks (IND-CCA1 and IND-CCA2) - Attacks under Message Non-malleability: NM-CPA and NM-CCA2 - Inter-relations among the attack model.*

UNIT III
Public key cryptography - RSA cryptosystem - probabilistic encryption - homomorphic encryption - Elliptic curve cryptosystems - Digital signatures and the notion of existential unforgability under chosen message attacks - ElGamal digital signature scheme - Schnorr signature scheme - blind signature.*

UNIT IV
Zero Knowledge Proofs and Protocols - Multi party computation: Models and definitions of Secure Computation - Secret Sharing Schemes - Oblivious Transfers (OT) and Extensions - Circuit Garbling.*

UNIT V
Blockchain technology - Consensus algorithm - Incentives and proof of work - Smart contract - Bitcoin.*

*Programming assignments are mandatory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Ability to understand the concepts of Blockchain Technology, Zero knowledge Proof and Multi party Computation
- Ability to break cryptosystems that are not provably secure
- Ability to derive simple provable security proofs for cryptographic schemes
- Ability to design and implement cryptographic protocols
- Ability to use cryptographic algorithms in security

Text Books

1. W. Mao, “Modern Cryptography: Theory & Practice”, Pearson Education, 2010.
2. Thomas Koshy, “Elementary Number Theory with Applications”, Elsevier India, 2005.
3. Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, **“**An Introduction to Mathematical Cryptography”, Springer Publication.

Reference Books

1. Menezes A et al., “Handbook of Applied Cryptography”, CRC Press, 1996.
2. Koblitz N., “Course on Number Theory and Cryptography”, Springer Verlag, 1986.

Mapping of Course Outcomes with Programme Outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	3	3	3	0	0	3	3	0	0	0	0	0
CO2	3	0	1	0	0	1	3	0	0	0	0	0
CO3	3	0	1	0	0	1	0	0	0	0	0	0
CO4	3	0	3	0	0	3	0	0	0	0	0	0
CO5	3	0	3	0	0	3	0	03	0	0	0	0

Course Code	:	**CSPE72**
Course Title	:	**Deep Learning Techniques**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC54**
Course Type	:	**PE**

Course Objectives

- To introduce building blocks of deep neural network architecture
- To learn deep learning algorithms and its problem settings
- To understand representation and transfer of knowledge using deep learning
- To learn to use deep learning tools and framework for solving real-life problems
- To use Python for Deep Learning

Course Contents

UNIT I Deep Networks
Deep Feedforward Networks - Learning XOR - Gradient Based learning - Hidden Units - Back-propagation and other Differential Algorithms - Regularization for Deep Learning - Optimization for training Deep Models.

UNIT II Convolutional Networks
Convolution operation - Motivation - Pooling - Convolution and Pooling as strong prior - Efficient convolution algorithms - Unsupervised features - Sequence Modeling: Recurrent and Recursive Nets - LSTM Networks - Applications - Computer Vision - Speech Recognition - Natural Language Processing.

UNIT III Linear factor Models
Probabilistic PCA and Factor Analysis - Independent Component Analysis (ICA) - Auto encoders - Regularized Auto encoders - Representational Power - Layer size and Depth - Stochastic Auto encoders - Applications.

UNIT IV Representation Learning
Greedy Layer-wise Unsupervised Pre-Training - Transfer learning and Domain Adaptation - Deep Generative Models.

UNIT V Deep Learning with Python
Introduction to Keras and Tensorflow - Deep Learning for computer vision - convnets - Deep Learning for Text and Sequences - Generative Deep Learning - Text Generation with LSTM - DeepDream - Neural Style Transfer - Generating images with variational autoencoders - Generative Adversarial Networks (GAN).

Course Outcomes

Upon completion of this course, the students will be able to:

- Identify the deep learning algorithms which are more appropriate for various types of learning tasks in various domains
- Incorporate transfer of knowledge in machine learning algorithms
- Implement deep learning algorithms and solve real-world problems
- Develop Deep Learning techniques using Python
- Represent learning Models

Text Book

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, The MIT Press, 2016.

Reference Books

1. Francois Chollet, “Deep Learning with Python”, Manning Publications, 2017.
2. Aurélien Géron, “Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”, First Edition, O'Reilly Media, 2017.
3. Josh Patterson, “Deep Learning: A Practitioner's Approach”, First Edition, O'Reilly Media.

Mapping of Course Outcomes with Programme Outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	0	2	3	0	0	3	0	0	0	0	3	0
CO2	2	3	3	0	0	0	0	0	1	0	1	0
CO3	3	0	3	0	0	3	0	0	0	0	0	3
CO4	3	0	0	3	0	3	0	0	0	1	2	0
CO5	2	3	0	0	0	3	0	0	0	0	3	0

Course Code	:	**CSPE73**
Course Title	:	**Natural Language Processing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC62**
Course Type	:	**PE**

Course Objectives

- To understand the steps involved in Natural language processing
- To learn about the lexical, syntactic and semantic analysis of natural language processing
- To explore the various parsing techniques for natural languages
- To understand the statistical models for Natural language processing
- To learn about the various applications involved in Natural language processing

Course Contents

UNIT I Lexical Analysis
Lexical Analysis - Regular expression and Automata for string matching - Words and Word Forms - Morphology fundamentals - Morphological Diversity of Indian Languages - Morphology Paradigms - Finite State Machine / Transducers Based Morphology - Automatic Morphology Learning - Parts of Speech - N-gram Models - Hidden Markov Models.*****

UNIT II Speech Processing
Biology of Speech Processing - Place and Manner of Articulation - Word Boundary Detection - Argmax based computations - HMM and Speech Recognition - Text to Speech Synthesis - Rule based-Concatenative based approach.*

UNIT III Parsing
Theories of Parsing - Parsing Algorithms – Earley Parser - CYK Parser - Probabilistic Parsing - CYK - Resolving attachment and structural ambiguity - Shallow Parsing - Dependency Parsing - Named Entity Recognition - Maximum Entropy Models - Conditional Random Fields.*

UNIT IV Lexical Knowledge Networks
Meaning: Lexical Knowledge Networks - Wordnet Theory - Indian Language Wordnets and Multilingual Dictionaries - Semantic Roles - Word Sense Disambiguation - WSD and Multilinguality - Metaphors - Coreference and Anaphora Resolution.*

UNIT V Applications
Applications: Sentiment Analysis - Text Entailment - Machine Translation - Question Answering System - Information Retrieval - Information Extraction - Cross Lingual Information Retrieval (CLIR).*

*Programming Assignments are mandatory.

Course Outcomes

Upon completion of the course, the students will be able to:

- Justify the various steps necessary for processing natural language
- Suggest appropriate lexical and parsing techniques for a given natural language
- Apply appropriate statistical models for a given natural language application
- Modify existing algorithms to suit any natural language for processing
- Suggest appropriate pre-processing steps essential for the various applications involving natural language processing

Text Books

1. Jurafsky Daniel, Martin James, “Speech and Language Processing”, Second Edition, Tenth Impression, Pearson Education, 2018.
2. Christopher Manning, Schutze Heinrich, “Foundations of Statistical Natural Language Processing”, MIT Press, 1999.

Reference Books

1. Allen James, “Natural Language Understanding”, Second Edition, Benjamin Cumming, 1995.
2. Charniack Eugene, “Statistical Language Learning”, MIT Press, 1993.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	1	0	0	1	0	0	0	0	1	0
CO2	0	2	0	0	3	0	0	0	3	3	0	0
CO3	3	0	2	0	0	3	0	0	0	0	0	3
CO4	0	3	3	0	2	0	0	0	3	0	2	0
CO5	0	1	1	0	1	1	0	0	1	0	0	1

Course Code	:	**CSPE74**
Course Title	:	**Image Processing and Applications**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To learn the fundamentals of image processing and various transformation applied in an image
- To learn image enhancement techniques
- To understand image restoration
- To impart knowledge on different compression techniques
- To discuss on image segmentation and feature representations

Course Contents

UNIT I Introduction

Introduction to Digital Image Processing - Characteristics of Digital Image - Basic relationship between pixels - Image sampling and quantization - Color models - Basic Geometric Transformations - Fourier Transform - Cosine-Sine and Hartley Transform - Hadamard-Haar-Slant Transform - Discrete Fourier Transform.

UNIT II Image Enhancement Techniques

Spatial Domain Methods - Basic Grey Level Transformation - Histogram Processing - Image subtraction - Image averaging - Spatial filtering - Smoothing - Sharpening filters - Laplacian filters - Frequency domain filters - Smoothing - Sharpening filters - Homomorphic filtering.

UNIT III Image Restoration

Model of Image Degradation/restoration process - Noise models - Spatial and Frequency Filters - Inverse filtering & Wiener Filtering - Least mean square filtering - Constrained least mean square filtering.

UNIT IV Image Compression Fundamentals

Image Compression Models - Lossless compression: Variable length coding - LZW coding - Bit plane coding - predictive coding - DPCM - Lossy Compression: Lossy Predictive Coding - Transform coding - Wavelet coding.

UNIT V Image Segmentation & Analysis

Image Segmentation techniques - Edge detection - Thresholding - Region - Boundary Extraction & Representation - Region - Moment representation - chain codes - Polygonal approximation - Texture - Pattern Recognition.
Applications - Finger print/iris recognition - Remote sensing - Automatic character recognition - Medical image processing.

Course Outcomes

Upon completion of this course, the students will be able to:

- Differentiate and interpret various image enhancement techniques
- Reconstruct the image from the degraded image
- Analyze and use appropriate image compression techniques
- Suggest proper image feature for classification problems
- Build image processing applications for real world problems

Text Book

1. Rafael C Gonzalez, Richard E Woods, “Digital Image Processing”, Fourth Edition, Pearson Education, 2018.

Reference Books

1. A.K. Jain, “Fundamentals of Digital Image Processing”, PHI, New Delhi, 1995.
2. William K Pratt, “Digital Image Processing”, Fourth Edition, John Wiley, 2007.
3. S E Umbaugh, “Digital Image Processing and Analysis: Application with MATLAB and CVIP Tools”, Third Edition , Taylor & Francis, CRC Press, 2018.
4. Frank Y. Shih, “Image Processing and Pattern Recognition”, Wiley – IEEE Press, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	3	2	0	2	0	0	0	0	2	0
CO2	3	0	2	0	1	0	0	0	0	0	0	2
CO3	2	3	2	0	1	2	0	0	0	0	2	0
CO4	3	3	0	0	0	3	0	0	0	0	2	0
CO5	2	0	3	3	1	3	0	0	0	0	2	1

Course Code	:	**CSPE75**
Course Title	:	**Network Security**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC63**
Course Type	:	**PE**

Course Objectives

- To understand the network security, services, attacks, mechanisms, types of attacks
- To comprehend and apply authentication services
- To apply authentication algorithms
- To comprehend and apply network layer security protocols, Transport layer security protocols, Web security protocols

Course Contents

UNIT I Overview of Network
Overview of Network Security - Security services - attacks - Security Issues in TCP/IP suite - Sniffing - spoofing - buffer overflow - ARP poisoning - ICMP Exploits - IP address spoofing - IP fragment attack - routing exploits - UDP exploits - TCP exploits.*

UNIT II Message Authentication Code
Authentication requirements - Authentication functions - Message Authentication Codes - Hash Functions - Security of Hash Functions and MACs - MD5 message Digest algorithm - Secure Hash Algorithm - RIPEMD - HMAC Digital Signatures - Authentication protocols - Kerberos - X.509.*

UNIT III IP Security
IP Security - AH and ESP - SSL/TLS - SSH - Web Security - HTTPS - DNS Security - Electronic Mail Security (PGP - S/MIME).*

UNIT IV Viruses
Intruders - Viruses - Worms - Trojan horses - Distributed Denial-Of-Service (DDoS) - Firewalls - IDS - Honey nets - Honey pots.*

UNIT V Introduction to Wireless Network Security
Introduction to wireless network security - Risks and Threats of Wireless networks - Wireless LAN Security (WEP - WPA).*

*Programming assignments are mandatory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Determine appropriate mechanisms for protecting the network
- Understand the security protocols and challenges
- Design and develop security solutions for a given application or system
- Apply Authentication algorithms for Security
- Ability to develop a secure network stack

Text Books

1. W. Stallings, “Cryptography and Network Security: Principles and Practice”, Fifth Edition, Prentice Hall, 2013.
2. Yang Xiao, Yi Pan, “Security in Distributed and Networking Systems”, World Scientific, 2007.

Reference Books

1. Aaron E. Earle, “Wireless Security Handbook”, Auerbach Publications, Taylor & Francis Group, 2006.
2. Atul Kahate, “Cryptography and Network Security”, Tata McGraw Hill, 2003.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	0	3	1	0	0	0	0	0	0
CO2	0	0	0	0	0	0	0	0	0	0	0	0
CO3	1	0	2	0	2	3	0	2	0	2	1	2
CO4	3	1	3	0	2	2	0	0	3	0	0	2
CO5	3	1	3	0	0	3	0	2	0	2	2	3

Course Code	:	**CSPE76**
Course Title	:	**Wireless Network Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC53**
Course Type	:	**PE**

Course Objectives

- Describe fundamentals of wireless communication
- Categorize architecture of different Wireless Networks
- Formulate significance of MAC and Network layers in Wireless Network System
- Differentiate Adhoc networks
- Relate MAC layer concepts

Course Contents

UNIT I Wireless Communications & Cellular System Fundamentals
Introduction to wireless communications systems - examples - comparisons and trends - Cellular systems - Frequency Management and Channel Assignment - types of handoff and their characteristics - dropped call rates & their evaluation - MAC techniques for Wireless Communication: FDMA - TDMA - MA (FHMA/ CDMA/ Hybrid techniques) - SDMA techniques.*

UNIT II Wireless WAN
Wireless WAN: First Generation Analog - Second Generation TDMA - GSM - Short Messaging Service in GSM - Second Generation CDMA–IS-95 - GPRS - Third Generation Systems (WCDMA/CDMA2000).*

UNIT III Wireless LAN Architecture
Wireless LAN: Introduction to wireless LANs - IEEE 802.11 WLAN - Architecture and Services - Physical Layer - MAC sub layer - MAC Management Sub layer - Other IEEE 802.11 standards -HIPERLAN -WiMAX standard.*

UNIT IV AdHoc Networks
Ad hoc and Sensor Networks: Characteristics of MANETs - Table-driven and Source-initiated On Demand routing protocols - Hybrid protocols - Wireless Sensor networks - Classification - MAC and Routing protocols.*

UNIT V MAC Layer
Wireless MAN and PAN: Wireless MANs - Physical and MAC layer details - Wireless PANs - Architecture of Bluetooth Systems - Physical and MAC layer details - Standards.*

*Programming assignments are mandatory.

Course Outcomes
Upon completion of this course, the students will be able to:

- Make critical assessment of wireless networks
- Comprehend the fundamentals of wireless networks
- Apply the knowledge gained in the development of MAC, network layer protocols of wireless network
- Categorize wireless sensor networks
- Collect the simulation tools for adhoc network

Text Books

1. William Stallings, “Wireless Communications and Networks”, Second Edition, Pearson/Prentice Hall of India, 2007.
2. Dharma Prakash Agrawal, Qing-An Zeng, “Introduction to Wireless and Mobile Systems”, Second Edition, Thomson India Edition, 2007.

Reference Books

1. C. Siva Ram Murthy, B. S. Manoj, “Adhoc Wireless Networks: Architectures and Protocols”, Second Edition, Pearson Education, 2008.
2. Jochen Schiller, “Mobile Communications”, Second Edition, Person Education, 2008.
3. Vijay K. Garg, “Wireless Communication and Networking”, Morgan Kaufmann Publishers, 2007.
4. Kaveth Pahlavan, Prashant Krishnamurthy, “Principles of Wireless Networks”, Pearson Education Asia, 2002.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	0	0	0	2	0	0	0	0	2	0
CO2	0	3	0	0	0	1	0	0	0	0	0	2
CO3	0	0	2	2	0	3	0	0	3	0	0	0
CO4	0	2	0	0	0	0	0	0	0	0	1	0
CO5	0	2	0	0	0	2	0	0	0	0	0	0

Course Code	:	**CSPE77**
Course Title	:	**Parallel Architectures and Programming**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC51**
Course Type	:	**PE**

Course Objectives

- To understand the fundamental principles involved in designing modern parallel computers
- To gain knowledge about key issues in parallel programming and architecture
- To understand the operation of parallel hardware including cache coherence and distributed memory machines
- To understand the necessity of efficient parallel program design to minimize overhead
- To gain knowledge about various parallel programming strategies
- To develop practical programming skills using MPI and CUDA

Course Contents

UNIT I Introduction

Introduction: The need for parallelism - Forms of parallelism (SISD - SIMD - MISD - MIMD) - Moore's Law and Multi-cores - Fundamentals of Parallel Computers - Communication architecture - Message passing architecture - Data parallel architecture - Dataflow architecture - Systolic architecture - Performance Issues.*

UNIT II Cache Design

Large Cache Design: Shared vs. Private Caches - Centralized vs. Distributed Shared Caches - Snooping-based cache coherence protocol - directory-based cache coherence protocol - Uniform Cache Access - Non-Uniform Cache Access - D-NUCA - S-NUCA - Inclusion - Exclusion - Difference between transaction and transactional memory - STM - HTM.*

UNIT III Graphics Processing Unit

Graphics Processing Unit: GPUs as Parallel Computers - Architecture of a modern GPU - Evolution of Graphics Pipelines - GPGPUs - Scalable GPUs - Architectural characteristics of Future Systems - Implication of Technology and Architecture for users - Vector addition - Applications of GPU.*

UNIT IV Parallel Programming

Introduction to Parallel Programming: Strategies - Mechanism - Performance theory - Parallel Programming Patterns: Nesting pattern - Parallel Control Pattern - Parallel Data Management - Map: Scaled Vector - Mandelbrot - Collative: Reduce - Fusing Map and Reduce - Scan - Fusing Map and Scan - Data Recognition: Gather - Scatter - Pack - Stencil and Recurrence - Fork-Join - Pipeline.*

UNIT V Parallel Programming Languages

Parallel Programming Languages: Distributed Memory Programming with MPI: trapezoidal rule in MPI - I/O handling - MPI derived data type - Collective Communication - Shared Memory Programming with Pthreads: Conditional Variables - read-write locks - Cache handling - Shared memory programming with Open MP: Parallel for directives - scheduling loops - Thread Safety - CUDA: Parallel programming in CUDA C - Thread management - Constant memory and Event - Graphics Interoperability - Atomics - Streams.*

*Programming assignments are mandatory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Comprehend parallel architecture and its importance in solving engineering problems
- Summarize and differentiate the different parallel programming strategies
- Design parallel programs to enhance machine performance in parallel hardware environment

- Design and write programs that can make efficient use of multiple cores, multiple networked processors and GPU Processing power
- Design and implement parallel programs in modern environments such as CUDA, OpenMP, etc.

Text Books

1. D. E. Culler, J. P. Singh, A. Gupta, “Parallel Computer Architecture”, Morgan-Kaufmann, 2004.
2. Rajeev Balasubramonian, Norman P. Jouppi, and Naveen Muralimanohar, “Multi-Core Cache Hierarchies”, Morgan & Claypool Publishers, 2011.
3. Peter, Pach Eco, “An Introduction to Parallel Programming”, Elsevier, 2011.
4. James R. Larus, Ravi Rajwar, “Transactional Memory”, Morgan & Claypool Publishers, 2007.

Reference Books

1. David B. Kirk, Wen-mei, W. Hwu, “Programming Massively Parallel Processors: A Hands-on Approach”, 2010.
2. Barbara Chapman, F. Desprez, Gerhard R. Joubert, Alain Lichnewsky, Frans Peters, “Parallel Computing: From Multicores and GPU's to Petascale”, 2010.
3. Michael McCool, James Reinders, Arch Robison, “Structured Parallel Programming: Patterns for Efficient Computation”, 2012.
4. Jason Sanders, Edward Kandrot, “CUDA by Example: An Introduction to General-Purpose GPU Programming”, 2011.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	0	0	3	2	0	0	0	0	3	0
CO2	0	3	0	0	0	0	0	0	0	0	3	0
CO3	3	0	3	0	2	3	2	0	3	0	3	0
CO4	0	3	3	0	0	3	0	0	0	0	0	3
CO5	0	3	2	0	3	2	0	0	3	0	0	0

Course Code	:	**CSPE78**
Course Title		**Information Security**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand the basics of Information Security
- To know the legal, ethical and professional issues in Information Security
- To know the aspects of risk management
- To become aware of various standards in this area
- To know the technological aspects of Information Security

Course Contents

UNIT I Introduction
History - What is Information Security? - Critical Characteristics of Information - NSTISSC Security Model - Components of an Information System - Securing the Components - Balancing Security and Access - The SDLC - The Security SDLC.

UNIT II Security Investigation
Need for Security - Business Needs - Threats - Attacks - Legal - Ethical and Professional Issues - An Overview of Computer Security - Access Control Matrix - Policy - Security policies - Confidentiality policies - Integrity policies and Hybrid policies.

UNIT III Security Analysis
Risk Management - Identifying and Assessing Risk - Assessing and Controlling Risk - Systems: Access Control Mechanisms - Information Flow and Confinement Problem.

UNIT IV Logical Design
Blueprint for Security - Information Security Policy - Standards and Practices - ISO 17799/BS 7799 - NIST Models - VISA - International Security Model - Design of Security Architecture - Planning for Continuity.

UNIT V Physical Design
Security Technology - IDS - Scanning and Analysis Tools - Cryptography - Access Control Devices - Physical Security - Security and Personnel.

Course Outcomes

Upon completion of this course, the students will be able to:

- Discuss the basics of information security
- Illustrate the legal, ethical and professional issues in information security
- Demonstrate the aspects of risk management
- Become aware of various standards in the Information Security System
- Design and implementation of Security Techniques

Text Book

1. Michael E Whitman, Herbert J Mattord, “Principles of Information Security”, Vikas Publishing House, New Delhi, 2003.

Reference Books

1. Micki Krause, Harold F. Tipton, “Handbook of Information Security Management”, Vol. 1-3, CRC Press LLC, 2004.
2. Stuart McClure, Joel Scrambray, George Kurtz, “Hacking Exposed”, Tata McGraw Hill, 2003.
3. Matt Bishop, “Computer Security Art and Science”, Pearson/PHI, 2002.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	1	0	0	1	0	0	0	0	1	0
CO2	3	3	0	0	0	0	0	1	0	1	1	0
CO3	0	0	0	0	0	0	0	3	0	2	2	0
CO4	0	0	0	0	0	0	0	2	0	0	2	0
CO5	1	0	2	0	2	3	0	0	3	0	3	0

Course Code	:	**CSPE79**
Course Title	:	**Human Computer Interaction**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To provide an overview of the concepts relating to the design of human-computer interfaces
- To understand the theoretical dimensions of human factors involved in the acceptance of computer interfaces
- To understand the important aspects of implementation of human-computer interfaces
- To identify the various tools and techniques for interface analysis, design and evaluation

Course Contents

UNIT I
HCI Foundations: Input–output channels - Human memory - Thinking: reasoning and problem solving - Emotion - Individual differences - Psychology and the design of interactive systems - Text entry devices - Positioning - pointing and drawing - Display devices - Devices for virtual reality and 3D interaction - Physical controls - sensors and special devices - Paper: printing and scanning.

UNIT II
Designing - Programming Interactive systems - Models of interaction - Frameworks and HCI - Ergonomics - Interaction styles - Elements of the WIMP interface - The context of the interaction - Experience - engagement and fun - Paradigms for interaction.
Cantered Design and testing - Interaction design basics - The process of design - User focus - Scenarios - Navigation design - Screen design and layout, Iteration and prototyping.

UNIT III
HCI in the software process - Iterative design and prototyping - Design rules - Principles to support usability - Standards and Guidelines - Golden rules and heuristics - HCI patterns.
Implementation support - Elements of windowing systems - Programming the application - Using toolkits - User interface management systems.

UNIT IV
Evaluation techniques - Evaluation through expert analysis - Evaluation through user participation - Universal design - User support.
Models and Theories - Cognitive models - Goal and task hierarchies - Linguistic models - The challenge of display-based systems - Physical and device models - Cognitive architectures.

UNIT V
Collaboration and communication - Face-to-face communication - Conversation - Text-based communication - Group working - Dialog design notations - Diagrammatic notations - Textual dialog notations - Dialog semantics - Dialog analysis and design Human factors and security - Groupware - Meeting and decision support systems - Shared applications and artifacts - Frameworks for groupware - Implementing synchronous groupware - Mixed - Augmented and Virtual Reality.

Course Outcomes

Upon completion of the course, the students will be able to:

- Design and Develop processes and life cycle of Human Computer Interaction
- Analyze product usability evaluations and testing methods
- Apply the interface design standards/guidelines for cross cultural and disabled users

- Categorize, Design and Develop Human Computer Interaction in proper architectural structures
- Understand various communication approaches

Text Books

1. A Dix, Janet Finlay, G D Abowd, R Beale, “Human - Computer Interaction’’, Third Edition, Pearson Publishers, 2008.
2. Shneiderman, Plaisant, Cohen, Jacobs, “Designing the User Interface: Strategies for Effective Human Computer Interaction”, Fifth Edition, Pearson Publishers, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	0	0	1	2	0	2	1	0	1	0
CO2	2	0	0	0	0	0	0	3	0	0	2	0
CO3	0	0	2	0	2	3	0	2	2	0	2	0
CO4	1	3	2	0	0	1	0	0	0	0	2	2
CO5	0	3	0	0	2	2	0	1	0	0	0	0

VIII SEMESTER

Course Code	:	**CSPE81**
Course Title	:	**Data Science**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand the data science process and exploration
- To learn Machine learning algorithms
- To get a knowledge on types of learning, processes, techniques and models
- To know about the research that requires the integration of large amounts of data

Course Contents

UNIT I

Introduction to data science - case for data science - data science classification - data science algorithms - Data Science Process - prior Knowledge - Data Preparation - Modeling - Application - Knowledge - Data Exploration - Objectives of data Exploration - Datasets - Descriptive Statistics - Data Visualization - Roadmap for data exploration.

UNIT II

Natural language Processing basics - Language Syntax and Structure - Language Semantics - Natural language Processing - Text Analytics - Text Preprocessing and Wrangling - Understanding Text Syntax and Structure - Feature Engineering for Text Representation - Traditional Feature Engineering Models - bag of words model - bag of N-Grams model - TF - IDF Model - Topic Models - Text Classification - Automated Text Classification - Text Classification Blueprint - Classification Models - Multinomial Naïve Bayes - Logistic Regression - Support Vector Machines - Ensemble Models - Random Forest - Gradient Boosting Machines - Evaluating Classification Models.

UNIT III

Text Similarity and clustering - Essential Concepts - Analyzing term Similarity - Analyzing Document Similarity - Document Clustering - Feature Engineering - K-means Clustering - Affinity Propagation - Ward's Agglomerative Hierarchical Clustering - Semantic Analysis - Exploring Wordnet - Word Sense Disambiguation - Named Entity Recognition - Analyzing Semantic Representations - Sentiment Analysis - Unsupervised Lexicon-Based Models - Bing Liu's Lexicon - MPQA Subjectivity Lexicon - Pattern Lexicon - TextBlob Lexicon - AFINN Lexicon - SentiWordNet Lexicon - VADER Lexicon - Classifying Sentiment with Supervised Learning.

UNIT IV

Speech - Phonetics - Speech Sounds and Phonetic Transcription - Articulatory Phonetics - Phonological Categories and Pronunciation variation - Acoustics Phonetics and Signals - Speech Synthesis - Phonetic Analysis - Prosodic Analysis - Diphone Waveform synthesis - Automatic Speech Recognition - Speech Recognition Architecture - Applying Hidden Markov Model to Speech - Feature Extraction: MFCC Vectors - Computing Acoustic Likelihoods - The Lexicon and language Model Search and decoding.

UNIT V

Time series Forecasting - Time series Decomposition - Smoothing based Methods - Regression based Methods - Machine Learning Methods - Performance evaluation - Anomaly Detection - Concepts - Distance based outlier Detection - Density based outlier Detection - Local outlier factor - Feature Selection - Classifying feature selection Methods - Principal Component Analysis - Information theory based filtering - chi-square based filtering - Wrapper-type feature selection.

Course Outcomes

Upon completion of this course, the students will be able to:

- Understand the data science concepts, techniques and models
- Forecast the time series data
- Build recommendation systems
- Learn and apply different mining algorithms and recommendation systems for large volumes of data
- Perform analytics on data streams

Text Books

1. Vijay Kotu, Bala Deshpande, "Data Science: Concepts and Practice", Second Edition, Elsevier Publications, 2019.
2. Brandon Reagen, Robert Adolf, Paul Whatmough, Gu-Yeon Wei, David Brooks, "Deep Learning for Computer Architects", Morgan Clay Pool Publishers, 2017.
3. Dipanjan Sarkar, "Text Analytics with Python: A Practitioner's Guide to Natural Language Processing", A Press, 2019.
4. Daniel Jurafsky, James H. Martin, "Speech and Language Processing", Pearson, 2009.

Reference Books

1. Ethem Alpaydin, "Introduction to Machine Learning", Third Edition, Adaptive Computation and Machine Learning Series, MIT Press, 2014.
2. Stephen Marsland, "Machine Learning – An Algorithmic Perspective", Second Edition, Machine Learning and Pattern Recognition Series, Chapman and Hall/CRC, 2014.
3. Dietmar Jannach, Markus Zanker, "Recommender Systems: An Introduction", Cambridge University Press, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	3	0	0	2	0	0	0	1	0	0	0
CO2	1	0	0	0	0	0	0	0	0	0	1	0
CO3	2	0	3	0	0	3	0	0	0	0	0	1
CO4	2	0	0	0	0	0	0	0	1	0	0	0
CO5	2	0	3	0	0	3	0	0	0	0	0	2

Course Code	:	**CSPE82**
Course Title	:	**GPU Computing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC51**
Course Type	:	**PE**

Course Objectives

- To learn the architecture of GPU and basics of parallelism
- To learn about the evolution of GPU computing
- To learn GPU programming using CUDA
- To study the methods of performance improvement in GPU

Course Contents

UNIT I Introduction
Graphics Processing Units (GPU) as Parallel Computers - Architecture of a modern GPU - Why more speed or parallelism? - Parallel Programming Languages and Models - Overarching Goals - History of GPU computing - Evolution of Graphics Pipelines - GPU Computing.

UNIT II Parallel Programming
Goals of Parallel Programming - Problem Decomposition - Algorithm Selection - Computational Thinking - Introduction to OPENCL: Background - Data Parallelism Model - Device Architecture - Kernel Functions - Device Management & Kernel Launch.

UNIT III Introduction to CUDA
Data Parallelism - CUDA Program Structure - AMatrix - Matrix Multiplication Example - Device Memories and Data Transfer - Kernel Functions and Threading - Function declarations - Kernel launch - Predefined variables - Runtime API - CUDA Threads: CUDA Thread Organization - Using blockIdx and threadIdx - Synchronization and Transparent Scalability - Thread Assignment - Thread Scheduling and Latency Tolerance - CUDA Memories: Importance of Memory Access Efficiency - CUDA Device Memory Types - A Strategy for Reducing Global Memory Traffic - Memory as a Limiting Factor to Parallelism.

UNIT IV Performance considerations
Thread execution - Global memory bandwidth - Dynamic partitioning of SM resources - Data prefetching - Instruction mix - Thread Granularity - Floating Point considerations: FP format - Representable numbers - Special bit patterns and precision - Arithmetic accuracy and rounding - Algorithm considerations - Debugging and Profiling: Debugging CUDA programs - Profiling CUDA programs - CUDA and MPI .

UNIT V Parallel Programming
Parallel Programming and Computational Thinking - Goals of Parallel Programming - Problem Decomposition - Algorithm Selection - Computational Thinking - OpenCL - Introduction.

Course Outcomes

Upon completion of this course, the students will be able to:

- Understand the basics of GPUs and GPU architecture
- Write programs for GPUs using CUDA and OpenCL
- Develop parallel applications targeting GPUs
- Develop Debugging tool
- Analyse the performance of the memory and thread execution in view of parallel programming

Text Book

1. David Kirk, Wen-mei Hwu, “Programming Massively Parallel Processors: A Hands-on Approach”, Third Edition, Morgan Kaufmann, 2017.

Reference Book

1. Shane Cook, “CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of GPU Computing)”, First Edition, Morgan Kaufmann, 2012.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	0	0	0	0	0	0	0	0	3	0
CO2	0	0	3	0	3	3	0	0	3	0	0	2
CO3	0	0	3	0	2	2	0	0	3	0	3	0
CO4	0	3	0	0	0	3	0	0	0	0	0	3
CO5	3	3	0	2	0	3	0	2	0	3	0	2

Course Code	:	**CSPE83**
Course Title	:	**Internet of Things – Principles and Practices**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand Smart Objects and IoT Architectures
- To learn about various IOT-related protocols
- To build simple IoT Systems using Arduino and Raspberry Pi.
- To understand data analytics and cloud in the context of IoT
- To develop IoT infrastructure for popular applications

Course Contents

UNIT I Fundamentals of IoT
Evolution of Internet of Things - Enabling Technologies - IoT Architectures: oneM2M - IoT World Forum (IoTWF) and Alternative IoT models - Simplified IoT Architecture and Core IoT Functional Stack - Fog - Edge and Cloud in IoT - Functional blocks of an IoT ecosystem - Sensors - Actuators - Smart Objects and Connecting Smart Objects.

UNIT II IoT Protocols
IoT Access Technologies: Physical and MAC layers - topology and Security of IEEE 802.15.4 - 802.15.4g - 802.15.4e - 1901.2a - 802.11ah and LoRaWAN - Network Layer: IP versions - Constrained Nodes and Constrained Networks – Optimizing IP for IoT: From 6LoWPAN to 6Lo - Routing over Low Power and Lossy Networks - Application Transport Methods: Supervisory Control and Data Acquisition - Application Layer Protocols: CoAP and MQTT.

UNIT III Design and Development
Design Methodology - Embedded computing logic - Microcontroller - System on Chips - IoT system building blocks - Arduino Board details - IDE programming - Raspberry Pi - Interfaces and Raspberry Pi with Python Programming.

UNIT IV Data Analytics and Supporting Services
Structured Vs Unstructured Data and Data in Motion Vs Data in Rest - Role of Machine Learning - No SQL Databases - Hadoop Ecosystem - Apache Kafka - Apache Spark - Edge Streaming Analytics and Network Analytics - Xively Cloud for IoT - Python Web Application Framework - Django AWS for IoT - System Management with NETCONF-YANG.

UNIT V Case Studies / Industrial Applications
Cisco IoT system - IBM Watson IoT platform – Manufacturing - Converged Plantwide Ethernet Model (CPwE) - Power Utility Industry - GridBlocks Reference Model - Smart and Connected Cities: Layered architecture - Smart Lighting - Smart Parking Architecture and Smart Traffic Control.

Course Outcomes

Upon completion of the course, the student should be able to:

- Explain the concept of IoT
- Analyze various protocols for IoT
- Design a PoC of an IoT system using Rasperry Pi/Arduino
- Apply data analytics and use cloud offerings related to IoT
- Analyze applications of IoT in real time scenario

Text Book

1. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Rob Barton, and Jerome Henry, “IoT Fundamentals: Networking Technologies, Protocols and Use Cases for Internet of Things”, Cisco Press, 2017.

Reference Books

1. Arshdeep Bahga, Vijay Madisetti, “Internet of Things – A Hands-on Approach”, Universities Press, 2015.
2. Olivier Hersent, David Boswarthick, Omar Elloumi, “The Internet of Things – Key Applications and Protocols”, Wiley, 2012 (for Unit 2).
3. Jan Holler et al., “From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence”, Elsevier, 2014.
4. Dieter Uckelmann, Mark Harrison, Michahelles, Florian (Eds), “Architecting the Internet of Things”, Springer, 2011.
5. Michael Margolis, “Arduino Cookbook, Recipes to Begin, Expand, and Enhance Your Projects”, Second Edition, O'Reilly Media, 2011.
6. *[Online]* https://www.arduino.cc/
7. *[Online]* https://www.ibm.com/smarterplanet/us/en/?ca=v_smarterplanet

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	2	0	0	1	0	0	0	1	1	0
CO2	2	3	0	0	0	0	0	0	0	1	2	0
CO3	1	0	3	0	3	0	0	0	2	0	0	0
CO4	3	0	1	0	0	1	0	0	0	0	0	2
CO5	0	0	3	0	0	1	0	0	0	0	2	0

Course Code	:	**CSPE84**
Course Title	:	**Social Network Analysis**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- Recognize the concept of semantic web and related applications
- Employ learn knowledge representation using ontology
- Recognize human behavior in social web and related communities
- Sketch and learn visualization of social networks
- Investigate variety of descriptive measures for networks and software to calculate them, and have the ability to interpret the results

Course Contents

UNIT I Introduction

Introduction to Semantic Web: Limitations of current Web - Development of Semantic Web - Emergence of the Social Web - Social Network analysis: Development of Social Network Analysis - Key concepts and measures in network analysis - Electronic sources for network analysis: Electronic discussion networks - Blogs and online communities - Web-based networks - Applications of Social Network Analysis.

UNIT II Modelling, Aggregating and Knowledge Presentation

Ontology and their role in the Semantic Web: Ontology-based knowledge Representation - Ontology languages for the Semantic Web: Resource Description Framework - Web Ontology Language - Modelling and aggregating social network data: State-of-the-art in network data representation - Ontological representation of social individuals - Ontological representation of social relationships - Aggregating and reasoning with social network data - Advanced representations.

UNIT III Extraction and Mining Communities in Web Social Networks

Extracting evolution of Web Community from a Series of Web Archive - Detecting communities in social networks - Definition of community - Evaluating communities - Methods for community detection and mining - Applications of community mining algorithms - Tools for detecting communities - social network infrastructures and communities - Decentralized online social networks - Multi-Relational characterization of dynamic social network communities.

UNIT IV Predicting Human Behavior and Privacy Issues

Understanding and predicting human behavior for social communities - User data management - Inference and Distribution - Enabling new human experiences - Reality mining - Context - Awareness - Privacy in online social networks - Trust in online environment - Trust models based on subjective logic - Trust network analysis - Trust transitivity analysis - Combining trust and reputation - Trust derivation based on trust comparisons - Attack spectrum and counter measures.

UNIT V Visualization and Applications of Social Networks

Graph theory - Centrality - Clustering - Node-Edge Diagrams - Matrix representation - Visualizing online social networks - Visualizing social networks with matrix-based representations - Matrix and Node-Link Diagrams - Hybrid representations - Applications - Cover networks - Community welfare - Collaboration networks - Co-Citation networks.

Course Outcomes

Upon completion of this course, the students will be able to:

- Develop semantic web related applications.
- Describe and Represent knowledge using ontology
- Inspect and Predict human behavior in social web and related communities

- Organize and Visualize social networks
- Analyze tools for detecting communities social network infrastructures

Text Books

1. Peter Mika, "Social Networks and the Semantic Web", First Edition, Springer 2007.
2. Borko Furht, "Handbook of Social Network Technologies and Applications", First Edition, Springer, 2010.

Reference Books

1. Guandong Xu, Yanchun Zhang, Lin Li, "Web Mining and Social Networking – Techniques and Applications", First Edition, Springer, 2011.
2. Dion Goh, Schubert Foo, "Social information Retrieval Systems: Emerging Technologies and Applications for Searching the Web Effectively", IGI Global Snippet, 2008.
3. Max Chevalier, Christine Julien, Chantal Soulé-Dupuy, "Collaborative and Social Information Retrieval and Access: Techniques for Improved User Modelling", IGI Global Snippet, 2009.
4. John G. Breslin, Alexander Passant, and Stefan Decker, "The Social Semantic Web", Springer, 2009.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	3	0	0	2	0	0	0	0	0	0
CO2	2	0	0	0	2	0	0	0	2	0	0	0
CO3	1	0	0	0	0	1	0	3	0	0	2	0
CO4	2	0	1	0	2	1	0	0	0	0	2	0
CO5	0	3	0	0	0	2	0	0	0	0	0	3

Course Code	:	**CSPE85**
Course Title	:	**Speech Processing Techniques**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To learn about the source of sound and its representation from a signal perspective
- To understand the features necessary for Speech processing
- To have an insight on the steps involved in Speech Recognition and Synthesis
- To learn about the mapping and features for identifying and extracting in music signal processing

Course Contents

UNIT I Speech Source and Representation
Speech Fundamentals - Articulatory Phonetics - Production and Classification of Speech Sounds - Acoustic Phonetics - acoustics of speech production - Review of Digital Signal Processing concepts - Short-Time Fourier Transform - Filter-Bank and LPC Methods.

UNIT II Speech Features
Speech Analysis - Features - Feature Extraction and Pattern Comparison Techniques: Speech distortion measures - mathematical and perceptual - Spectral Distance - Cepstral Distances - LPC - PLP and MFCC Coefficients - Time Alignment and Normalization - Dynamic Time Warping - Multiple Time - Alignment Paths.

UNIT III Speech Recognition
Speech Modeling - Hidden Markov Models: Markov Processes - HMMs - Evaluation - Optimal State Sequence - Viterbi Search - Baum-Welch Parameter Re-estimation - Implementation issues - Speech Recognition: Large Vocabulary Continuous Speech Recognition: Architecture of a large vocabulary continuous speech recognition system - acoustics and language models - N-grams - context dependent sub-word units - Applications and present status - Speech Recognition - Framework.

UNIT IV Speech Synthesis
Speech Synthesis - Text-to-Speech Synthesis - Concatenative and waveform synthesis methods - subword units for TTS - intelligibility and naturalness - role of prosody - Applications and present status - Speech synthesis framework - Case study.

UNIT V Applications
Speaker Recognition - Verification - voice biometrics - music processing - issues - representation - Pitch - Melody - Timbre - Music Features - Singer identification - Instrument identification.

Course Outcomes

Upon completion of the course, the students will be able to:

- Understanding basics of speech and signal processing
- Appreciate the Time, Spectral and Cepstral features' influence in recognizing and synthesising speech
- Decide and design the appropriate combination of features for applications involving speech processing and recognition
- Propose new strategies and approaches for language independent speech synthesis and recognition
- Decide on combination of signal features for music processing

Text Books

1. Lawrence Rabiner, Biing-Hwang Juang, "Fundamentals of Speech Recognition", Pearson Education, 2003.
2. Daniel Jurafsky, James H Martin, "Speech and Language Processing – An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", Pearson Education, 2013.

Reference Books

1. Steven W. Smith, “The Scientist and Engineer’s Guide to Digital Signal Processing”, California Technical Publishing.
2. Thomas F Quatieri, “Discrete-Time Speech Signal Processing – Principles and Practice”, Pearson Education.
3. Claudio Becchetti, Lucio PrinaRicotti, “Speech Recognition”, John Wiley and Sons, 1999.
4. Ben Gold, Nelson Morgan, “Speech and Audio Signal Processing, Processing and Perception of Speech and Music”, Wiley India Edition.
5. Frederick Jelinek, “Statistical Methods of Speech Recognition”, MIT Press, 1997.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	0	0	2	0	0	1	0	0	0	0
CO2	3	0	0	0	0	2	0	0	0	0	0	0
CO3	0	3	2	0	3	2	0	0	2	0	2	0
CO4	3	3	2	0	0	2	0	0	0	0	0	1
CO5	3	3	0	0	0	2	0	0	0	0	2	0

Course Code	:	**CSPE86**
Course Title	:	**Deep Learning Paradigms for Computer Vision**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Courses Objectives:

- To understand the basic ideas and principles of Computer Vision
- To understand the motion analysis Techniques in Computer Vision
- To understand and implement Deep Learning Architectures
- To understand deep learning models for computer vision
- To understand the methods of solving real life problems with respect to computer Vision using deep learning techniques.

Course Content:

UNIT I BASICS OF COMPUTER VISION
Introduction-Image Formation-Image Representation-Linear Filtering-Image in frequency domain- Image Sampling-Edge Detection-Feature detection - SIFT and its variants- Image Segmentation-Feature matching

UNIT II MOTION ANALYSIS
Background Subtraction and Modeling-Optical Flow- KLT- Spatio-Temporal Analysis-Dynamic Stereo- Motion parameter estimation

UNIT III DEEP LEARNING ARCHITECTURES
CNN Architectures – Convolution – Pooling Layers – Back propagation in CNN-Transfer Learning -RNN, LSTM, GRU, Encoder/Decoder Architectures – Autoencoders –Variational Autoencoders – Adversarial Generative Networks – Self Attention Mechanism

UNIT IV DEEP LEARNING MODELS FOR COMPUTER VISION
Object Classification-VGGNET, RESNET, ALEXNET, DENSENET, EFFICIENT NET, MOBILENET, INCEPTION V3, Object Detection-R-CNN, F-RCN, SSD, Retinanet, YOLO, CornerNet, Image Segmentation- U-Net, SegNet, Mask-RCNN, Attention Models-Transformers

UNIT V APPLICATIONS AND RECENT TRENDS IN COMPUTER VISION
Applications- Image Editing, Inpainting, Superresolution, 3D Object Generation, Security, Surveillance-Object Tracking-Automatic Image Captioning.
Recent Trends- Zero-shot, One-shot, Few-shot Learning-Self-supervised Learning and Reinforcement Learning in Vision

Course Outcomes:

After the completion of the course, the students will be able to
1. Implement fundamental image processing techniques required for computer vision
2. Employ the motion analysis techniques for solving real life problem
3. Apply the deep learning architectures to various problems
4. Create their own deep learning models
5. Develop applications of computer vision using deep learning techniques

Text books:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, 2016.
2. Richard Szeliski, Computer Vision: Algorithms and Applications, 2010.
3. Simon Prince, Computer Vision: Models, Learning, and Inference, 2012.

Reference Books:

1. Yoshua Bengio, Learning Deep Architectures for AI, 2009.
2. Michael Nielsen, Neural Networks and Deep Learning, 2016.
3. David Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2002.
4. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Second Edition, Cambridge University Press, March 2004.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	0	0	3	0	0	0	0	0	0	0
CO2	3	3	0	0	1	0	0	0	0	1	0	0
CO3	2	2	0	0	1	0	0	0	0	0	0	0
CO4	3	0	0	0	0	3	0	0	0	0	0	0
CO5	0	3	0	0	2	0	1	0	0	0	0	0

MINORS

Course Code	:	**CSMI11**
Course Title	:	**Data Structures and Algorithms**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- To understand the various techniques of sorting and searching
- To design and implement arrays, stacks, queues, and linked lists
- To understand the complex data structures such as trees and graphs
- To design and implement various programming paradigms and its complexity

Course Contents

UNIT I Development of Algorithms
Notations and analysis - Storage structures for arrays - Sparse matrices - Stacks and Queues: Representations and applications - linked lists - Doubly linked lists - Circular linked lists.

UNIT II Trees
Preliminaries - Binary Trees - Search Tree ADT - Binary Search Trees - Hashing: ADT Hash Function - Separate Chaining - Open Addressing - Rehashing - Extendible Hashing.

UNIT III Graphs
Representation of graphs - BFS - DFS - Topological sort - Shortest path problems - Dijkstra's algorithm - Floyd-Warshall - Minimum spanning trees - prims algorithm - Kruskal algorithm.

Unit IV Algorithmic Paradigms
Divide and Conquer method - Strassen's matrix multiplication - Greedy method - Knapsack problem - Job sequencing with deadlines - Dynamic Programming - Travelling salesman problem.

UNIT V Searching and Sorting Techniques
Selection - Bubble - Insertion - Merge - Quick - Radix sort - Address calculation - Linear search - Binary search.

Course Outcomes

Upon completion of this course, the students will be able to:

- Develop programs to implement linear data structures such as stacks, queues, linked lists, etc
- Apply the concept of trees, hashing algorithms and analysing their performance
- Utilize the suitable graph algorithms in different real world scenarios
- Solve optimization problems with the suitable algorithmic design techniques
- Comprehend the implementation of sorting and searching algorithms

Text Book

1. J. P. Tremblay, P. G. Sorenson, “An Introduction to Data Structures with Applications”, Second Edition, Tata McGraw Hill, 1981.

Reference Books

1. SartajSahni, “Data Structures, Algorithms and Applications in C++”, Universities Press Pvt. Ltd.
2. T. Cormen, C. Lieserson, R. Rivest, C. Stein, “Introductions to Algorithms”, Third Edition, Prentice-Hall/India, 2009.
3. M. Tenenbaum, Augestien, “Data Structures using C”, Third Edition, Pearson Education, 2007.

Mapping of Course Outcomes with Programme Outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**
CO1	1	1	1	2	2	3	0	0	0	0	0	0
CO2	0	1	2	0	0	2	0	0	3	0	3	0
CO3	1	0	1	2	0	0	0	0	0	3	0	0
CO4	0	2	1	0	2	0	0	3	0	0	3	0
CO5	1	0	2	0	0	3	0	0	0	0	0	0

Course Code	:	**CSMI12**
Course Title	:	**Computer Organization**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- To understand the basic hardware and software issues of computer organization
- To understand how computations are performed at machine level
- To understand how data storage is happening at machine level
- To understand the memory hierarchies, cache memories and virtual memories
- To learn the different ways of communication with I/O devices

Course Contents

UNIT I
Introduction - Technologies for building Processors and Memory - Performance - The Power Wall operations of the Computer Hardware - Operands - Signed and Unsigned numbers - Representing Instructions - Logical Operations - Instructions for Making Decisions.

UNIT II
MIPS Addressing for 32-Bit Immediate Addresses - Parallelism and Instructions: Synchronization - Translating and Starting a Program - Addition and Subtraction - Multiplication - Division - Floating Point - Parallelism and Computer Arithmetic: Subword Parallelism - Streaming SIMD Extensions.

UNIT III
Logic Design Conventions - Building a Datapath - A Simple Implementation Scheme - overview of Pipelining - Pipelined Datapath - Data Hazards: Forwarding versus Stalling - Control Hazards - Exceptions - Parallelism via Instructions.

UNIT IV
Memory Technologies - Basics of Caches - Measuring and Improving Cache Performance - Dependable memory hierarchy - Virtual Machines - Virtual Memory - Using FSM to Control a Simple Cache - Parallelism and Memory Hierarchy: Redundant Arrays of Inexpensive Disks.

UNIT V
Disk Storage and Dependability - Parallelism and Memory Hierarchy: RAID levels - Performance of storage systems - Introduction to multithreading clusters - message passing multiprocessors.

Course Outcomes

Upon completion of the course, the students will be able to:

- Understand the architecture and functionality of central processing unit
- Analyze the abstraction of various components of a computer
- Analyze the hardware and software issues and the interfacing
- Work out the tradeoffs involved in designing a modern computer system
- Understand the various memory systems and I/O communication

Text Books

1. David A. Patterson, John L. Hennessey, “Computer Organization and Design, The Hardware/Software Interface”, Fifth Edition, Morgan Kauffman/Elsevier, 2014.
2. Smruti Ranjan Sarangi, “Computer Organization and Architecture”, McGraw Hill Education, 2015.

Reference Books

1. V. Carl Hamacher, Zvonko G. Varanesic, Safat G. Zaky, "Computer Organization", Sixth Edition, McGraw Hill Inc., 2012.
2. William Stallings, "Computer Organization and Architecture", Eighth Edition, Pearson Education, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	2	0	0	0	2	0	0	0	0	0
CO2	0	0	2	0	0	0	3	0	0	0	0	0
CO3	0	0	3	0	2	0	0	0	0	0	0	0
CO4	3	3	0	0	1	0	2	0	0	2	0	3
CO5	0	0	0	0	3	0	1	0	0	0	0	0

Course Code	:	**CSMI13**
Course Title	:	**Operating Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- To provide basic knowledge about the services rendered by operating systems
- To explain the various issues related to process management
- To provide a detailed discussion of the various memory management techniques
- To discuss the various file-system design and implementation issues
- To discuss how the protection domains, help to achieve security in a system

Course Contents

UNIT I
Basic OS Concepts - User's view of the OS - Architectural support - OS services - OS structures - System calls - Building and Booting OS - Process - Threads - Multithreading.

UNIT II
Thread and process scheduling - Types of schedulers - Scheduling Policies – Inter-process synchronization - Critical Section problem - Hardware and Software solutions.

UNIT III
Semaphores - Monitors – Inter-process communication - Deadlocks: Characterization - Handling of deadlocks - Prevention - Avoidance - detection and recovery.

UNIT IV
Memory Management - Contiguous allocation - Static and dynamic partitioned memory allocation - Non-contiguous allocation - Paging - Segmentation - Virtual Memory - Demand Paging.

UNIT V
Need for files - File abstraction - File naming - File system organization - File system optimization - Reliability - Security and protection - I/O management and disk scheduling - Recent trends and development.
Case Study: Linux and Windows OS

Course Outcomes

Upon completion of the course, the students will be able to:

- Comprehend the techniques used to implement the process manager
- Comprehend virtual memory abstractions in operating systems
- Design and develop file system interfaces
- Design protection mechanisms for securing the system
- Design and manage storage systems

Text Book

1. Silberschatz, Galvin, Gagne, “Operating System Concepts”, Tenth Edition, John Wiley and Sons, 2018.

References Books

1. William Stallings, “Operating Systems – Internals and Design Principles”, Eighth Edition, Pearson Publications, 2014.
2. Andrew S. Tanenbaum, “Modern Operating Systems”, Fourth Edition, Pearson Publications, 2014.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	2	2	0	0	0	0	0	0	0	0
CO2	2	2	0	0	0	0	0	0	0	0	0	0
CO3	2	0	3	3	2	3	0	1	2	0	2	1
CO4	2	2	2	2	2	3	0	1	2	0	2	1
CO5	0	0	0	2	1	1	1	1	1	1	2	1

Course Code	:	**CSMI14**
Course Title	:	**Database Management Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- To learn data models, conceptualize and depict a database system using ER diagram
- To understand the internal storage structures in a physical DB design
- To know the fundamental concepts of transaction processing techniques
- To understand the concept of Database Design in Normalization techniques
- To know the manipulation of SQL Queries

Course Contents

UNIT I Introduction
Purpose of Database System - Views of data - Data Models - Database Languages - Database System Architecture - components of DBMS - Entity - Relationship model (E-R model) - E-R Diagram notation - Examples.

UNIT II Relational Model
Relational Data Model - Concept of relations - schema-instance distinction - keys - integrity rules - relational algebra operators - SQL: data definition - data manipulation - aggregate function - Null Values - nested sub queries - Joined relations - Work with MySQL Workbench.

UNIT III Database Design
Dependencies and Normal forms - dependency theory - functional dependencies - Armstrong's axioms for FD's - closure of a set of FD's - minimal covers - definitions of 1NF - 2NF - 3NF and BCNF - 4NF - 5NF, decompositions and desirable properties of them.

UNIT IV Transaction management
ACID properties - serializability and concurrency control - Lock based concurrency control (2PL - Deadlocks) - database recovery management.

UNIT V Implementation Techniques
Overview of Physical Storage Media - Magnetic Disks - RAID - Tertiary storage - Organization of Records in Files - Indexing and Hashing - Ordered Indices - primary - secondary index structures - Static Hashing - Dynamic Hashing.

Course Outcomes

Upon completion of this course, the students will be able to:

- Install, configure, and interact with a relational database management system
- Master the basics of SQL and construct queries using SQL
- Design and develop a large database with optimal query processing
- Develop efficient storage scheme of saving and retrieving Records and Files
- Design the database with normalization techniques

Text Books

1. Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Fifth Edition, Tata McGraw Hill, 2006.
2. J. Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.

Reference Books

1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Fourth Edition, Pearson/Addision Wesley, 2007.
2. Raghu Ramakrishnan, “Database Management Systems”, Third Edition, McGraw Hill, 2003.
3. S. K. Singh, “Database Systems Concepts, Design and Applications”, First Edition, Pearson Education, 2006.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	3	0	0	2	0	0	0	0	0	0
CO2	1	0	3	0	0	2	0	0	2	0	2	0
CO3	2	0	3	0	3	0	0	0	3	0	1	0
CO4	1	0	3	0	0	2	0	0	0	0	0	2
CO5	1	0	3	0	0	1	0	0	0	0	0	2

Course Code	:	**CSMI15**
Course Title	:	**Software Engineering**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- Memorize Software Engineering Practice& Process Models
- Identify requirements prioritization
- Construct data and Architectural Design diagram
- Differentiate software testing methods
- Collect ideas about software project management and Maintenance

Course Contents

UNIT I Software Engineering Modeling
The Evolving role of Software - The changing Nature of Software - Legacy software - A generic view of process - A layered Technology - A Process Framework - Software Development Life cycle (SDLC) Models: WaterFall Model - Prototype Model - Spiral Model - Evolutionary Development Models - Iterative Enhancement Models - Incremental Process Model - Concurrent Development Model - The Unified Process.

UNIT II Requirement analysis
Software Requirements - Functional & non-functional - Software requirements document - Requirement engineering process: Feasibility studies - Elicitation - Validation & Management - Software prototyping - Analysis and modelling.

UNIT III Software Design Concepts
Design Concepts and Principles - Modular design - Design heuristic - S/W architecture - Data design - Architectural design - Transform & Transaction mapping - Introduction to SCM process - Software Configuration Items.

UNIT IV Software Testing Strategy
Software Testing - Taxonomy of S/W testing - Black box testing - Testing boundary conditions - Structural testing - Regression testing - S/W testing strategies: Unit testing - Integration testing - Validation testing - System testing and debugging.

UNIT V Project Management and Maintenance
Software Project Management - S/W cost estimation - Function point models - COCOMO model - Project Scheduling - S/W maintenance.

Course Outcomes
Upon completion of this course, the students will be able to:

- Enhance the software project management skills
- Comprehend the systematic methodologies involved in SE
- Design and develop a software product in accordance with SE principles
- Organize all the testing tools and its usage
- Collect Software cost estimation tools

Text Books

1. Roger S. Pressman, “Software Engineering: A Practitioner’s Approach”, Seventh Edition, McGraw Hill International Edition, 2009.
2. Ian Sommerville, “Software Engineering”, Ninth Edition, Pearson Education, 2011.

Reference Books

1. Pankaj Jalote, “An Integrated Approach to software Engineering”, Springer Verlag, 1997.
2. Pfleeger, and Lawrence, “Software Engineering: Theory and Practice”, Second Edition, Pearson Education, 2001.
3. Bennatan, Edwin M., “On Time within Budget: Software Project Management Practices and Techniques”, John Wiley & Sons Inc., 1992.
4. McConnell, Steve, “Software Estimation: Demystifying the Black Art”, Microsoft Press, 2006.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	0	0	2	0	0	3	1	2	1	0
CO2	2	0	0	0	2	0	0	0	3	0	0	0
CO3	3	0	3	0	0	2	0	0	0	0	3	0
CO4	9	3	1	0	0	1	0	0	0	0	2	0
CO5	1	2	1	0	0	1	0	0	0	0	2	0

Course Code	:	**CSMI16**
Course Title	:	**Computer Networks**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- To provide insight about fundamental concepts and reference models (OSI and TCP/IP) and its functionalists
- To gain comprehensive knowledge about the principles, protocols, and significance of Layers in OSI and TCP/IP
- To know the implementation of various protocols and cryptography techniques
- Learn the flow control and congestion control algorithms

Course Contents

UNIT I Data Communications
Data Transmission - Multiplexing - Data Encoding Techniques - Introduction to computer networks - Network - Topologies - Reference Models: ISO/OSI Model and TCP/IP Model.

UNIT II Physical Layer
Transmission Media - Analog signals - Digital Signals - Data Link Layer - Error Detection and Correction - Parity - LRC - CRC - Hamming Code - Flow Control and Error Control - Stop and wait - ARQ - Sliding window - IEEE 802.3 Ethernet.

UNIT III Network Layer
Packet Switching and Circuit Switching - IP addressing methods - Subnetting - Supernetting - Routing Protocols: IP - ARP - RARP - DHCP - Routing Algorithms: Distance Vector Routing - Link State Routing.

UNIT IV Transport Layer
Transport Services - UDP - TCP - Congestion Control - Quality of Services (QOS).

UNIT V Application Layer
Domain Name Space (DNS) - Electronic Mail - HTTP - WWW.

Course Outcomes

Upon completion of this course, the students will be able to:

- Gain insight about basic network theory and layered communication architectures
- Identify the components required to build different types of networks
- Choose the required functionality at each layer for given application
- Identify solution for each functionality at each layer
- Trace the flow of information from one node to another node in the network

Text Books

1. Andrew S. Tanenbaum, David J. Wetherall, “Computer Networks”, Fifth Edition, Prentice Hall, 2011.
2. Behrouz A. Foruzan, “Data Communication and Networking”, Fifth Edition, Science Engineering & Math Publications, 2013.

Reference Books

1. W. Stallings, “Data and Computer Communication”, Tenth Edition, Pearson Education, 2014.
2. Larry L. Peterson, Bruce S. Davie, “Computer Networks: A Systems Approach”, Fifth Edition, Morgan Kaufmann Publishers, 2011.
3. Nader. F. Mir, “Computer and Communication Networks”, Pearson Prentice Hall Publishers, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	0	0	0	0	0	0	0	0	0	0
CO2	1	0	0	0	2	1	0	0	3	0	0	0
CO3	2	0	0	0	2	1	0	0	3	0	3	0
CO4	0	2	0	0	1	1	0	0	3	0	3	0
CO5	0	0	0	0	1	0	0	0	0	0	1	0

Course Code	:	**CSMI17**
Course Title	:	**Artificial Intelligence**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- To understand the various characteristics of Intelligent agents
- To learn the different search strategies in AI
- To learn to represent knowledge in solving AI problems
- To understand the ways of planning and acting in the real world
- To know about the models behind the AI application

Course Contents

UNIT I Introduction

Introduction - Definition - Future of Artificial Intelligence - Characteristics of Intelligent Agents - Typical Intelligent Agents - Problem Solving Approach to Typical AI problems.

UNIT II Problem Solving Methods

Problem solving Methods - Search Strategies - Uninformed - Informed - Heuristics - Local Search Algorithms and Optimization Problems - Searching with Partial Observations - Backtracking Search - Performance of search algorithms.

UNIT III Knowledge Representation

First Order Predicate Logic - Unification - Forward Chaining - Backward Chaining - Resolution - Knowledge Representation using First order Predicate logic - Reasoning Systems.

UNIT IV Planning

Planning with state-space search - partial-order planning - planning graphs - planning and acting in the real world - Plan generation systems.

UNIT V Uncertain Knowledge and Reasoning

Uncertainty - review of probability - probabilistic Reasoning - Bayesian networks - inferences in Bayesian networks - Temporal models - Hidden Markov models.

Course Outcomes

- Ability to design a plan for the real world problems and mapping it to the digital world
- Suggest appropriate search strategies for any AI problem
- Design agents for any given problem
- Appreciate the uncertainty in designing AI systems and propose algorithms for the same
- Ability to identify problems that are amenably solved by AI methods

Text Book

1. S. Russel, P. Norvig, “Artificial Intelligence – A Modern Approach”, Third Edition, Pearson Education, 2015.

Reference Books

1. Kevin Night, Elaine Rich, Nair B., “Artificial Intelligence (SIE)”, Third Edition, McGraw Hill, 2017.
2. Dan W. Patterson, “Introduction to AI and ES”, Pearson Education, 2007.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	3	3	2	3	0	0	0	0	3	0
CO2	3	3	0	1	0	0	0	0	0	0	0	0
CO3	3	3	0	2	0	0	0	0	0	0	0	0
CO4	3	3	0	1	0	0	0	0	0	0	0	0
CO5	0	2	0	3	0	3	0	0	2	0	3	2

HONOURS

Course Code	:	**CSHO11**
Course Title	:	**Software Defined Networking**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**MI**

Course Objectives

- Analyze reduced Complexity of Network Operation
- Describe and understand the concepts of minimize Layer and maximize Network Resources
- Evaluate and understand the Faster Time to Revenue for New Applications
- Memorize Data center and its usage
- Illustrate about Big data

Course Contents

UNIT I Introduction

Introduction - Control Plane - Data Plane - Distributed Control Planes - IP and MPLS - Creating the IP Underlay - Convergence Time - Load Balancing High Availability - Creating the MPLS Overlay - Replication - Centralized Control Planes – Logical Versus Litera - ATM/LANE - Route Servers - Wire Protocol - FAWG - Config and Extensibility - Architecture - Hybrid Approaches - Ships in the Night - Dual Function Switches.*

UNIT II Interface

VMWare - Nicira - Mininet - NOX/POX - Trema - Ryu - Big Switch Networks/Floodlight - Layer 3 Centric - L3VPN - Path Computation Element Server - Plexxi Affinity - Cisco OnePK - Management Interface - Network Divide - Modern Programmatic Interfaces - Modern Orchestration.*

UNIT III Data Center

Multitenant Data Center - Virtualized Multitenant Data Center - SDN Solutions for Data Center Network - VLANs - EVPN - VxLan - NVGRE - Virtualization and Data Plane I/O - Services Engineered Path - Service Locations and Chaining - NEV at ETSI - Non-ETSI NEV Work.*

UNIT IV Topology

Network Topology - Traditional Methods - LLDP - BGP-TE/LS - ALTO - I2RS - Build Code First - The Juniper SDN Framework(s) - Open Daylight Controller/Framework - Policy.*

UNIT V Technology

Bandwidth Scheduling - Manipulation - Calendaring - Bandwidth Calendaring - Big Data and Application Hyper - Virtualization for Instant CSPF - Expanding Technology - Use Cases for Data Center Overlays - Big Data - Network Function Virtualization - Data Center Orchestration - Puppet - Network Function Virtualization - Optimized Big Data - Firewall as Service - Network Access Control Replacement - Virtual Firewall - Feed Back and Optimization - Intrusion Detection/Threat Mitigation.*

*Programming Assignments are mandatory

Course Outcomes

Upon completion of this course, the students will be able to:

- Comprehend Software Defined Networks
- Compare and analyze the advantages of SDN over traditional network
- Design and implement software defined network
- Design algorithm for virtualization
- Design algorithm for big data analytics

Text Books

1. Thomas D. Nandeau, Ken Gray, “Software Defined Networks”, First Edition, O’ Reilly Media Inc., 2013.
2. FEI HU, “Network Innovation through OpenFlow and SDN: Principles and Design”, CRC Press, Taylor & Francis Group, 2014.

Reference Books

1. Azodolmolky, Siamak, “Software Defined Networking with OpenFlow”, Packt Publishing Ltd., 2013.
2. Nadeau, Thomas D., Ken Gray, “SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies”, O'Reilly Media Inc., 2013.
3. Dillinger, Markus, Kambiz Madani, Nancy Alonistioti, “Software Defined Radio: Architectures, Systems and Functions”, John Wiley & Sons, 2005.
4. Goransson, Paul, Chuck Black, Timothy Culver, “Software Defined Networks: A Comprehensive Approach”, Morgan Kaufmann, 2016.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	3	0	2	0	1	0	0	0	0	1	0
CO2	0	2	1	0	0	1	0	0	0	0	0	0
CO3	2	0	2	0	2	3	0	0	3	0	3	0
CO4	1	1	2	0	2	3	0	0	2	0	0	3
CO5	1	1	2	0	1	1	0	0	3	0	0	3

Course Code	:	**CSHO12**
Course Title	:	**Distributed Systems**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**CSPC41**
Course Type	:	**HO**

Course Objectives

- To understand fundamentals of Distributed Systems
- To explore the issues in communications in distributed systems
- To understand the various issues in process and thread management
- To discuss the issues in design the distributed file system
- To know security issues in Distributed System

Course Contents

UNIT I Introduction
Parallel and Distributed Systems - multiprocessor versus multicomputer systems - Message-passing systems versus shared memory systems - Primitives for distributed communication - Synchronous versus asynchronous executions - Design issues and challenges - Distributed Computing paradigms.

UNIT II Procedures and Methods
Issues in communication - Remote Procedure Call - Remote Method Invocation - Message- and Stream-Oriented communication.

UNIT III Synchronization
Processes and threads - Code migration and distributed scheduling - Naming - Clock Synchronization - Distributed mutual exclusion and distributed deadlocks.

UNIT IV Transaction
Distributed transaction - Consistency models - Replication - Fault tolerance - Distributed commit and failure recovery - Distributed file systems (NFS - AFS & coda).

UNIT V Security
Security in distributed systems - Security: authentication - Distributed middleware: CORBA - Case studies: DCOM and JINI.

Course Outcomes

Upon completion of this course, the students will be able to:

- Apply the middleware technologies in designing a distributed system
- Apply remote method invocation and objects
- Design process and resource management systems
- Handle faults in real time environments
- Handle security related issues in Distributed System

Text Book

1. Coulouris, Dollimore, Kindberg, “Distributed Systems - Concepts and Design”, Pearson Education Asia.

Reference Books

1. Kshemkalyani, Ajay D., Mukesh Singhal, “Distributed Computing: Principles, Algorithms, and Systems”, Cambridge University Press, 2011.
2. P K Sinha, “Distibuted Operating System”, PHI, IEEE Press.
3. Singhal, Shivaratri, “Advanced Concepts in Operating Systems”, TMH.
4. Tanenbaum, “Distributed Systems: Principles and Paradigms”, Pearson Education.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	3	0	3	3	0	0	3	0	3	1
CO2	0	3	0	3	2	0	0	0	3	0	0	0
CO3	0	0	3	0	3	0	0	0	2	0	3	0
CO4	0	0	2	0	0	2	0	0	3	0	3	3
CO5	3	0	3	0	0	3	2	0	0	0	3	3

Course Code	:	**CSHO13**
Course Title	:	**Multi-Core Programming**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**CSPC51**
Course Type	:	**HONOURS**

Course Objectives

- To understand the fundamentals of multi-core architecture
- To be able to know the basic concepts of multi core programming using threads
- To be able to understand various programming constructs in multi-core architecture
- To be able to understand Multithreaded applications

Course Contents

UNIT I Introduction to Multiprocessors and Scalability Issues
Scalable design principles - Principles of processor design - Instruction Level Parallelism - Thread level parallelism - Parallel computer models - Symmetric and distributed shared memory architectures - Performance Issues - Multi-core Architectures - Software and hardware multithreading - SMT and CMP architectures - Design issues - Case studies - Intel Multi-core architecture - SUN CMP architecture**.**

UNIT II Parallel Programming
Fundamental concepts - Designing for threads - scheduling - Threading and parallel programming constructs - Synchronization - Critical sections - Deadlock - Threading APIs.

UNIT III OpenMP Programming
OpenMP - Threading a loop - Thread overheads - Performance issues - Library functions - Solutions to parallel programming problems - Data races - deadlocks and livelocks - Non-blocking algorithms - Memory and cache related issues.

UNIT IV MPI Programming
MPI Model - collective communication - data decomposition - communicators and topologies - point-to-point communication - MPI Library.

UNIT V Multi Threaded Application Development
Algorithms - program development and performance tuning.

Course Outcomes

Upon completion of this course, the students will be able to:

- To understand the limitations of ILP and the necessity of multi-core architecture
- To be able to know the basic concepts of multi core programming to manage threads using the role of OpenMP
- To be able to understand various programming constructs and Solve the issues related to multiprocessing and suggest solutions in multicore architecture
- To be able to understand Multithreaded applications
- To design scalable and high-performance software systems, which includes performance analysis, algorithmic techniques for high performance, instruction-level optimizations, caching optimizations, parallel programming, and building scalable systems

Text Books

1. Shameem Akhter, Jason Roberts, “Multi-core Programming”, Intel Press, 2006.
2. Michael J Quinn, “Parallel programming in C with MPI and OpenMP”, Tata McGraw Hill, 2003.

Reference Books

1. John L. Hennessey, David A. Patterson, “Computer Architecture – A Quantitative Approach”, Fourth Edition, Morgan Kaufmann/Elsevier Publishers, 2007.
2. David E. Culler, Jaswinder Pal Singh, “Parallel Computing Architecture: A Hardware/Software Approach”, Morgan Kaufmann/Elsevier Publishers, 1999.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	0	3	3	0	0	0	0	0	0	1
CO2	3	2	0	2	0	0	0	3	3	0	0	1
CO3	3	0	0	3	3	0	0	3	0	0	0	3
CO4	0	2	0	3	3	0	0	3	0	0	0	3
CO5	3	3	0	2	0	3	2	0	0	0	0	2

Course Code	:	**CSHO14**
Course Title	:	**Pervasive and Ubiquitous Computing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**CSPC43**
Course Type	:	**HO**

Course Objectives

- To understand the characteristics and principles of Pervasive computing
- To understand the various components that helps to build pervasive computing system
- To understand the necessity of sensor networks and RFID that capture and disseminate context information
- To understand the principles, challenges, infrastructures and user interface that supports the ubiquitous computing
- To design and implement Pervasive and Ubiquitous applications that are embedded in everyday objects

Course Contents

UNIT I Introduction

Pervasive Computing: Principles - Characteristics - interaction transparency - context aware - automated experience capture - Vision and challenges of pervasive computing - Pervasive computing infrastructure - Architecture for pervasive computing - Pervasive devices - embedded controls - smart sensors and actuators -Context communication and access services.

UNIT II Technologies

Device Technology for Pervasive Computing: Hardware - Human-machine interfaces - Biometrics - Operating Systems - Java for pervasive devices - Voice Technology: Basics of Speech Recognition - Voice standards - Speech Applications - Speech and Pervasive Computing - Security - Personal Digital Assistants.

UNIT III Sensor Networks and RFID

Introduction to Sensor networks: Sensor Node Architecture - Sensor Network Architecture - Types of sensor networks - Platforms for Wireless sensor networks - Applications of Wireless Sensor networks - Introduction to RFID - transponder and reader architecture - Types of tags and readers - Frequencies of operation - Application of RFID Technologies.

UNIT IV Introduction to Ubiquitous Computing

An introduction - overview - challenges to research topics in ubiquitous computing including sensors - ambient displays - tangibles - middleware - mobility - allocation and context awareness - Architecture for ubiquitous computing: new devices and communications - software architectures - Wireless standards & protocols for ubiquitous networks - Near field communication (NFC) - Bluetooth classic - Bluetooth Low Energy (BLE) - WiFi - WiFi Direct.

UNIT V Ubiquitous Computing Applications

Ubiquitous applications: the appropriate design - Weiser's vision of ubiquitous computing - mixed reality and sensible design - Wearable computing - Glass and Augmented Reality - Eye-Tracking-Digital Pen and Paper Mobile social networking & crowd sensing Event based social network.

Course Outcomes

Upon completion of this course, the students will be able to:

- Understand the fundamental theoretical concepts in pervasive computing
- Conclude the enabling technologies that drive the pervasive and ubiquitous computing
- Analyze and compare the performance of different data dissemination techniques
- Formulate the design aspects, that are essential to create the model of pervasive computing
- Develop solutions for problems related to pervasive and ubiquitous computing system through investigation

Text Books

1. Jochen Burkhardt, Horst Henn, Stefan Hepper, Thomas Schaec, Klaus Rindtorff, “Pervasive Computing: Technology and Architecture of Mobile Internet Applications”, Sixth Edition, Pearson Education, New Delhi, 2009.
2. Seng Loke, “Context-Aware Computing Pervasive Systems”, Auerbach Pub., Taylor and Francis Group, New York, 2007.
3. John Krumm, “Ubiquitous Computing Fundamentals”, CRC Press, 2010.

Reference Books

1. Rahul Banerjee, “Lecture Notes in Pervasive Computing”, Outline Notes, BITS-Pilani, 2012.
2. Genco, S. Sorce, “Pervasive Systems and Ubiquitous Computing”, WIT Press, 2012.
3. Guruduth S. Banavar, Norman H. Cohen, Chandra Narayanaswami, “Pervasive Computing: An Application-Based Approach”, Wiley Interscience, 2012.
4. Frank Adelstein, S K S Gupta, G G Richard, L Schwiebert, “Fundamentals of Mobile and Pervasive Computing”, Tata McGraw Hill, New Delhi, 2005.
5. Stefen Poslad, “Ubiquitous Computing: Smart Devices, Environments and Interactions”, Second Edition, Wiley, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	1	0	0	0	2	0	0	0	0	0
CO2	2	0	3	0	0	3	0	0	0	0	2	0
CO3	0	2	0	3	3	2	0	0	1	0	0	0
CO4	2	0	2	0	0	3	0	0	0	0	3	0
CO5	0	3	3	0	0	2	0	0	0	0	1	3

Course Code	:	**CSHO15**
Course Title	:	**Middleware Technologies**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	-
Course Type	:	**HO**

Course Objectives

- To understand the essence of client-server and middleware architectures
- To understand the concepts distributed applications
- To learn the basics of CORBA and C#.NET technologies
- To learn the concepts of java bean

Course Contents

Unit I Introduction
Introduction to client server computing - client server models - Benefits of client server computing - pitfalls of client server programming – Middleware - Client/server building blocks - RPC - RMI.*

Unit II Middleware
Middleware - Objects - Elements - Architecture - Middleware distributed applications - middleware types - transaction oriented middleware.*

Unit III Corba
CORBA with Java - Client/Server CORBA-style - CORBA with Java-Static - CORBA ORBlets with Applets - Dynamic CORBA Beans - CORBA initialization protocol - CORBA activation services - CORBA java-to-IDL mapping.*

Unit IV EJB
EJBs and CORBA - Object transaction monitors - CORBA OTM's - EJB and CORBA OTM's - EJB container framework - Session and Entity Beans - EJB client/server development Process - The EJB container protocol - support for transaction EJB packaging - EJB design Guidelines.*

Unit V Object Oriented Programming
C# and .NET Platform - .NET Assemblies - Object Oriented Programming with C# - Callback Interfaces - Delegates - Events - Type Reflection - Late Binding - Attribute-Based Programming - Object Serialization and the .NET Remoting Layer.*

*Programming assignments are mandatory.

Course Outcomes
Upon completion of this course, the students will be able to:

- Comprehend of Middleware tools
- Build real time applications based on .Net and C#
- Design, develop, and analyze middleware architecture in developing enterprise technologies
- Handle EJB tools
- Design real time remote applications using C#

Text Books

1. Andrew Troelsen, "C# and the .NET Platform", Second Edition, Apress Wiley-Dreamtech, India Pvt. Ltd., 2007.
2. Chris Britton, "IT Architectures and Middleware: Strategies for Building Large, Integrated Systems", Second Edition, Pearson Education, 2004.

Reference Books

1. Robert Orfali, Dan Harkey, Jeri Edwards, “The Essential Client/Server Survival Guide”, Third Edition, John Wiley & Sons, 1999.
2. Robert Orfali, Dan Harkey, “Client/Server Programming with Java and CORBA”, Second Edition, John Wiley & Sons, SPD, 1998.
3. Jesse Liberty, “Programming C#”, Second Edition, O’Reilly Press, 2002.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	3	0	0	2	0	0	0	0	0	0
CO2	0	0	3	3	0	3	0	0	2	0	2	0
CO3	0	2	3	3	0	3	0	0	0	0	0	1
CO4	0	0	1	0	0	1	0	0	0	0	2	0
CO5	0	0	3	0	0	3	0	0	3	0	1	0

Course Code	:	**CSHO16**
Course Title	:	**Randomized Algorithms**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**CSPC42**
Course Type	:	**HO**

Course Objectives

- To introduce the concept of randomized algorithms
- To apply the concepts of probabilistic analysis of algorithms
- To derive good upper bounds for the expected running time of simple randomized algorithm
- To analyze the performance of randomized algorithms
- To design simple randomized algorithm that run faster or return correct output with high probability

Course Contents

UNIT I Probability and Computing

Elements of probability theory - Verification of strings - poly identities - matrix multiplication - Las Vegas and Monte Carlo algorithms - Expectations - Jensen's Inequality - Coupon collector's problem - geometric distribution.*

UNIT II Expectations, Moments and Inequalities

Randomized Quick Sort and its expected run-time - Variance and moments - Chebyshev's inequality - Coupon collector's problem - randomized median finding - analysis - moment generating functions.*

UNIT III Chernoff's Bounds and its Applications

Derivation and application of Chernoff's bounds - Sum of Poisson Trials - Coin flips - Set balancing - Packet routing in sparse networks - permutation routing on the hypercube - butterfly.*

UNIT IV Balls, Bins and Random Graphs

Birthday paradox - balls and bins model - application to bucket sort - Poisson distribution - Application to hashing - random graph models - Hamiltonian cycles in random graphs.*

UNIT V Markov Chains and Random Walks

Markov chains - representations - randomized algorithm for 2-satisfiability and 3-satisfiability - classification of states - gambler's ruin - random walks on undirected graphs - s-t connectivity algorithm.*

*Programming assignments are mandatory.

Course Outcomes

Upon completion of this course, the students will be able to:

- Learn the mathematical foundations emphasizing the design and analysis of randomized algorithm
- Apply basics of probability theory in the analysis of algorithms
- Comprehend randomized algorithms and its advantages to traditional algorithm
- Design and implement randomized techniques in solving real world problems
- Design randomized algorithms and analyze their performance

Text Book

1. M. Mitzenmacher, E. Upfal, "Probability and Computing: Randomized Algorithms and Probabilistic Analysis", Cambridge, 2005.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	0	3	2	0	0	0	0	0	0
CO2	1	0	2	0	2	0	0	0	0	0	2	0
CO3	0	1	0	0	2	0	0	0	3	0	1	0
CO4	3	0	3	0	1	2	0	0	0	0	1	0
CO5	3	0	2	0	3	2	0	0	2	0	2	0

Course Code	:	**CSHO17**
Course Title	:	**Big Data Mining**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**HO**

Course Objectives

- To understand big data and data analytics lifecycle
- To learn Basic Data analytic methods using R
- To Get a knowledge on advanced analytical methods, technology and tools

Course Contents

UNIT I Big Data Overview
State of the practice in Analytics - Key roles for new big data ecosystem - Data Analytics Lifecycle - Data analytics lifecycle overview - Discovery - Data Preparation - Model Planning - Model Building - Communicate Results - operationalize.

UNIT II Introduction to R
Exploratory Data Analytics - Statistical methods for evaluation - Hadoop & Map Reduce framework for R - R with Relational Database Management Systems - R with Non-Relational (NoSQL) DBs.

UNIT III Clustering
Overview of Clustering - K-means - Association Rules – Overview - Apriori Algorithm - Evaluation of candidate rules - An Example: Transactions in grocery Store - Validation and Testing - Diagnostics - Regression - Linear Regression - Logistic Regression - Reason to choose and Cautions - Additional Regression Models.

UNIT IV Classification
Decision Trees - Naïve Bayes - Diagnostics of Classifiers - Additional classification methods - Time series Analysis - Overview of Time series analysis - ARIMA Model - Additional methods - Text Analysis - Text analysis steps - A text analysis Example - Collecting raw Text - Representing Text-Term Frequency - Inverse document frequency (TFIDF) - Categorizing documents by Topics - Determining Sentiments - Gaining insights.

UNIT V Analytics for Unstructured data
The Hadoop Ecosystem – NoSQL - In-Database Analytics - SQL Essentials - In-Database Text Analysis - Advanced SQL.

Course Outcomes

Upon completion of this course, the students will be able to:

- Understand the big data concepts
- Utilize and apply the Analytical methods, Technology and tools in the industry
- Understand hadoop ecosystem and apply to solve real-life problems
- Understand the principles and concepts of machine learning
- Learn the usage of machine learning techniques to find the hidden patterns in big data

Text Books

1. EMC Education Services, “Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data”, Wiley publishers, 2015.
2. Simon Walkowiak, “Big Data Analytics with R”, PackT Publishers, 2016.

Reference Books

1. Bart Baesens, “Analytics in a Big Data World: The Essential Guide to Data Science and its Applications”, Wiley Publishers, 2015.
2. Kim H. Pries, Robert Dunnigan, “Big Data Analytics: A Practical Guide for Managers”, CRC Press, 2015.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	2	0	0	3	0	0	0	1	0	0	0
CO2	2	0	2	2	0	3	0	0	0	0	0	0
CO3	0	0	0	0	3	0	0	0	3	0	0	0
CO4	3	0	2	0	0	2	0	0	0	0	0	0
CO5	2	1	3	0	0	0	0	0	0	0	1	0

Course Code	:	**CSHO18**
Course Title	:	**Modelling and Simulation**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**HO**

Course Objectives

- To define important terminologies and classify systems/models
- To understand continuous and discrete modelling of computer systems
- To understand the analytical modelling of computer systems
- To enable the students to develop new queueing analysis for both simple and complex systems
- To appreciate the use of smart scheduling and introduce the students to analytical techniques for evaluating scheduling policies

Course Contents

UNIT I Simulation
Inventory Concept: The technique of Simulation - Major application areas - concept of a System - Continuous and discrete systems - Systems modeling - types of models - Progress of a Simulation Study - Monte Carlo Method - Comparison of Simulation and Analytical Methods.

UNIT II Applications
Discrete-Time Markov Chains - Ergodicity Theory - Real World Examples - Google - Aloha - Transition to Continuous-Time Markov Chain - M/M/1 and PASTA.

UNIT III Queueing Analysis
Server Farms: M/M/k and M/M/k/k - Capacity Provisioning for Server Farms - Time Reversibility and Burke's Theorem - Networks of Queues and Jackson Product Form - Classed and Closed Networks of Queues.

UNIT IV Matrix Analytic Methods
Case Study of Real-world Workloads - Phase-Type Distributions and Matrix-Analytic Methods - Networks with Time-Sharing Servers - M/G/1 Queue and the Inspection Paradox - Task Assignment Policies for Server Farms.

UNIT V Scheduling Policies
Performance Metrics - Scheduling - Non-Preemptive and Preemptive Non-Size-Based Policies - Scheduling Non-Preemptive and Preemptive Size-Based Policies - Scheduling - SRPT and Fairness.

Course Outcomes

Upon completion of this course, the students will be able to:

- Identify the technique for discrete and continuous models
- Discuss open and closed queueing networks
- Apply the operational laws to open and closed systems
- Use discrete-time and continuous-time Markov chains to model real world systems
- Develop analytical techniques for evaluating scheduling policies

Text Books

1. G. Gordon, "System Simulation", Second Edition, Prentice Hall, 2002.
2. Mor Harchol-Balter, "Performance Modeling and Design of Computer Systems – Queueing Theory in Action", Cambridge University Press, 2013.

Reference Books

1. Raj Jain, “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling”, Wiley-Interscience, 1991.
2. Lieven Eeckhout, “Computer Architecture Performance Evaluation Methods”, Morgan and Claypool Publishers, 2010.
3. A. M. Law, W. D. Kelton, “Simulation Modelling and Analysis”, Fifth Edition, McGraw Hill, 2014.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	0	0	1	0	0	0	0	0	1	0
CO2	3	3	0	0	2	0	0	0	1	0	1	0
CO3	1	0	2	0	0	2	0	1	0	0	1	0
CO4	1	0	2	0	0	2	0	0	3	0	0	3
CO5	1	0	1	0	2	3	0	0	3	0	3	0

Course Code	:	**CSHO19**
Course Title	:	**Quantum Computing**
Number of Credits	:	**3-1-0-4**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**HO**

Course Objectives

• To understand the principles of quantum computation and mechanics
• To learn about the operators involved in Quantum computing and their applications
• To study the information theory aspects of quantum computing
• To explore the various error corrections available for quantum computing
• To comprehend the applications of quantum computing to information theory and cryptography

Course Contents

UNIT I Introduction to Quantum Computation
Global perspectives - Quantum bits - Quantum computation - Quantum algorithms - Classical computations on a quantum computer, Quantum parallelism - Deutsch's algorithm - Deutsch–Jozsa algorithm - Quantum algorithms summarized - Experimental quantum information processing, Quantum information - Postulates of quantum mechanics - Application: superdense coding - Density operator - Schmidt decomposition and purifications - EPR and the Bell inequality

UNIT II Quantum Computing
Operations Quantum circuits - Single qubit operations - Controlled operations - Measurement - Universal quantum gates - Simulation of quantum systems, Quantum Fourier transform and its applications - Phase estimation - Applications: order-finding and factoring - General applications of the quantum Fourier - Quantum search algorithms - Quantum search as a quantum simulation, Quantum counting - Speeding up the solution of NP-complete problems - Quantum search of an unstructured database - Optimality of the search algorithm - Black box algorithm limits

UNIT III Quantum Information
Quantum computers: physical realization - Guiding principles - Conditions for quantum computation- Harmonic oscillator quantum computer - Optical photon quantum computer - Optical cavity quantum electrodynamics - Ion traps, Nuclear magnetic resonance - Other implementation schemes - Quantum information - Quantum noise and quantum operations - Classical noise and Markov processes, Quantum operations Examples of quantum noise and quantum operations - Applications of quantum operations - Limitations of the quantum operations formalism

UNIT IV Quantum Error Correction
Distance measures for quantum information - Distance measures for classical information- closeness of two quantum states – wellness of quantum channel information preservation- Quantum error-correction- Three qubit bit flip code - Three qubit phase flip code - Shor code, Theory of quantum error-correction- Constructing quantum codes - Classical linear codes – Calderbank–Shor–Steane codes- Stabilizer codes - Stabilizer formalism - Unitary gates and the stabilizer formalism - Measurement in the stabilizer formalism - Gottesman–Knill theorem - Stabilizer code constructions, Fault-tolerant quantum computation- Fault-tolerance: the big picture- Fault-tolerant quantum logic- Fault-tolerant measurement- Elements of resilient quantum computation

UNIT V Quantum Information Theory
Entropy and information - Shannon entropy- Basic properties of entropy - Von Neumann entropy- Strong subadditivity, Quantum information theory- Distinguishing quantum states and the accessible information – Data compression, Classical information over noisy quantum channels - Quantum information over noisy quantum channels - Entanglement as a physical resource, Quantum cryptography

Course Outcomes

Upon completion of this course, the students will be able to:
• To understand the principles of quantum computation and mechanics
• To learn about the operators involved in Quantum computing and their applications

• To study the information theory aspects of quantum computing
• To explore the various error corrections available for quantum computing
• To comprehend the applications of quantum computing to information theory and cryptography

Text Books

1. Nielsen, M., & Chuang, I. (2010). In Quantum Computation and Quantum Information: 10th Anniversary Edition, Cambridge: Cambridge University Press.
2. John Gribbin (2014), Computing with Quantum Cats: From Colossus to Qubits, Prometheus Books.
3. B.N. Murdin (2013) Quantum Computing from the Ground Up, by Riley Tipton Perry, Contemporary Physics

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	2	0	0	1	0	0	0	0	0	0
CO2	3	0	2	2	0	0	0	0	0	0	0	0
CO3	0	2	2	0	0	2	0	0	0	0	0	0
CO4	3	2	2	0	0	3	0	0	0	0	0	0
CO5	0	0	3	0	0	0	2	0	0	0	0	0

OPEN ELECTIVES

Course Code	:	**CSOE11**
Course Title	:	**Big Data Analytics**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand the Big Data Platform and its Use cases
- To Provide an overview of Apache Hadoop
- To Provide HDFS Concepts and Interfacing with HDFS
- To understand NoSQL database

Course Contents

UNIT I Introduction to Big Data
Introduction: Big Data - Characteristics of Big Data - Big data management architecture - Examining Big Data Types - Big Data Technology Components - Big data analytics - Big data analytics examples - Web Data Overview - Web Data in Action.

UNIT II Hadoop
Introduction: History of Hadoop - Hadoop Ecosystem - Analyzing data with Hadoop - Hadoop Distributed File System - Design - HDFS concepts - Hadoop filesystem - Data flow - Hadoop I / O - Data integrity - Serialization - Setting up a Hadoop cluster - Cluster specification - cluster setup and installation - YARN.

UNIT III MapReduce
Introduction: Understanding MapReduce functions - Scaling out - Anatomy of a MapReduce Job Run - Failures - Shuffle and sort - MapReduce types and formats - features - counters - sorting - MapReduce Applications – Configuring and setting the environment - Unit test with MR unit - local test.

UNIT IV Spark
Installing spark - Spark applications - Jobs - Stages and Tasks - Resilient Distributed databases - Anatomy of a Spark Job Run - Spark on YARN - SCALA: Introduction - Classes and objects - Basic types and operators - built-in control structures - functions and closures - inheritance.

UNIT V NoSQL Databases
Introduction to NoSQL - MongoDB: Introduction - Data types - Creating - Updating and deleing documents - Querying - Introduction to indexing - Capped collections - Hbase: Concepts - Hbase Vs RDBMS - Creating records - Accessing data - Updating and deleting data - Modifying data - exporting and importing data.
USE CASES: Call detail log analysis - Credit fraud alert - Weather forecast.

Course Outcomes

Upon completion of this course, the students will be able to:

- Understand the characteristics of big data and concepts of Hadoop ecosystem
- Understand the concepts of Scala programming
- Apply Mapreduce programming model to process big data
- Analyze Spark and its uses for big data processing
- Design programs for big data applications using Hadoop components

Text Books

1. EMC Education Services, “Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data”, Wiley Publishers, 2015.
2. Simon Walkowiak, “Big Data Analytics with R”, PackT Publishers, 2016.

Reference Books

1. David Loshin, “Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, No SQL, and Graph”, Morgan Kaufmann/Elsevier Publishers, 2013.
2. Bart Baesens, “Analytics in a Big Data World: The Essential Guide to Data Science and its Applications”, Wiley Publishers, 2015.
3. Kim H. Pries, Robert Dunnigan, “Big Data Analytics: A Practical Guide for Managers”, CRC Press, 2015.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	1	0	0	3	0	0	0	0	3	0
CO2	0	1	0	1	0	0	0	0	1	0	1	0
CO3	3	0	2	0	1	2	0	0	3	0	0	1
CO4	3	1	0	0	2	0	0	0	3	0	0	0
CO5	0	1	1	0	2	2	0	0	2	0	0	2

Course Code	:	**CSOE12**
Course Title	:	**Cloud & Grid Computing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**OE**

Course Objectives

- To provide comprehensive knowledge of fundamental concepts and of grid and cloud computing
- To demonstrate an understanding of Virtualization, Service models and deployment models of the cloud
- To describe the programming and software environments of grid and cloud
- To shed light on the security issues in the grid and the cloud

Course Contents

UNIT I
Overview of Distributed Computing - Cluster Computing - Technologies for Network based systems - Software environments for Distributed Systems - Overview of Services and Service oriented Architecture.

UNIT II
Fundamentals of Grid Computing - Open Grid Services Architecture - Motivation - Functionality Requirements - Practical & Detailed view of OGSA/OGSI - Data intensive grid service models - OGSA services.

UNIT III
Virtual Machines and Virtualization - Implementation levels of Virtualization - Virtualization structures/tools and Mechanisms - Virtualization of CPU - Memory and I/O Devices - Storage Virtualization.

UNIT IV
Cloud Computing - Properties - challenges - Service models - IaaS - PaaS and SaaS Deployment models - Service Composition and orchestration - Architecture design of Compute and Storage cloud - Public Cloud Platforms - Inter Cloud Resource Management.

UNIT V
Grid Security Issues - The Grid Security Infrastructure - Authorization modes in GSI - Possible Vulnerabilities - Cloud security issues - Infrastructure security - Data security - Identity and access management Privacy - Audit and Compliance.

Course Outcomes

Upon completion of this course, the students will be able to:

- Understand the core concepts of Distributed computing
- Articulate the Virtualization concepts
- Identify the architecture, service models and deployment models of Cloud
- Master the programming aspects of the Grid and Cloud
- Understand and build secure and reliable Grid and Cloud applications

Text Books

1. Kai Hwang, Geoffrey C. Fox, Jack J. Dongarra, “Distributed and Cloud Computing from Parallel Processing to the Internet of Things”, Morgan Kaufmann, Elsevier, 2012.
2. Frédéric Magoulès, “Fundamentals of Grid Computing Theory, Algorithms and Technologies”, CRC Press, 2010.

Reference Books

1. Ian T. Foster et al., "The Grid 2: Blueprint for a New Computing Infrastructure", Elsevier, 2003.
2. Erl, Thomas, Ricardo Puttini, Zaigham Mahmood, "Cloud Computing: Concepts, Technology & Architecture", Pearson Education, 2013.
3. Hurwitz, Judith S., Robin Bloor, Marcia Kaufman, Fern Halper, "Cloud Computing for Dummies", John Wiley & Sons, 2010.
4. Maozhen Li, Mark Baker, "The Grid – Core Technologies", John Wiley & Sons, 2005.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	1	0	0	0	2	0	0	0	0	0
CO2	2	0	2	0	0	3	0	0	0	0	0	0
CO3	0	3	1	0	0	2	0	0	0	0	2	0
CO4	0	2	0	0	3	3	0	0	2	0	1	0
CO5	0	2	0	0	0	2	3	0	0	0	3	2

Course Code	:	**CSOE13**
Course Title	:	**Computer Graphics and Multimedia Systems**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**OE**

Course Objectives

- To understand the basics of various inputs and output computer graphics hardware devices
- Exploration of fundamental concepts in 2D and 3D computer graphics
- To know 2D raster graphics techniques, 3D modeling, geometric transformations, 3D viewing and rendering
- Exploration of fundamental concepts in multimedia systems, file handling, hypermedia

Course Contents

UNIT I
Basic of Computer Graphics: Applications of computer graphics - Display devices - Random and Raster scan systems - color models - Graphics Primitives: Points - lines - circles and ellipses as primitives - scan conversion algorithms for primitives.*

UNIT II
Two-Dimensional Graphics: Two dimensional geometric transformations - Matrix representations and homogeneous coordinates - composite transformations - Two dimensional viewing - viewing pipeline - viewing coordinate reference frame - window-to-viewport coordinate transformation - Two dimensional viewing functions - clipping operations - point - line - polygon clipping algorithms.*

UNIT III
Three-Dimensional Graphics: Three dimensional concepts - Three dimensional object representations - Polygon surfaces - Polygon tables - Plane equations - Polygon meshes - Curved Lines and surfaces - Quadratic surfaces - Blobby objects - Spline representations - Bezier curves and surfaces - B-Spline curves and surfaces - Transformation and Viewing: Three dimensional geometric and modeling transformations - Translation - Rotation - Scaling - composite transformations - Three dimensional viewing – viewing pipeline - viewing coordinates - Projections - Clipping.*

UNIT IV
Multimedia System Design & Multimedia File Handling: Data and File Formats - Multimedia basics - Multimedia applications - Multimedia system architecture - Evolving technologies for multimedia - Defining objects for multimedia systems - Multimedia data interface standards - Multimedia databases - Compression and decompression - Data and file format standards - Multimedia I/O technologies - Digital voice and audio - Video image and animation - Full motion video - Storage and retrieval technologies.*

UNIT – V
Hypermedia: Multimedia authoring and user interface - Hypermedia messaging - Mobile messaging - Hypermedia message component - Creating hypermedia message - Integrated multimedia message standards - Integrated document management - Distributed multimedia systems.*
*Programming assignments are mandatory.

Course Outcomes

Upon completion of the course, the students will be able to:

- Understand the various computer graphics hardware and display technologies
- Implement various 2D and 3D objects transformation techniques
- Apply 2D and 3D viewing technologies into the real world applications
- Implement multimedia components efficiently
- Understanding Hypermedia and distributed multimedia systems

Text Books

1. J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes, “Computer Graphics: Principles and practice”, Second Edition in C, Addison Wesley, 1997.
2. Donald Hearn, Pauline Baker M, “Computer Graphics”, Prentice Hall, New Delhi, 2007.
3. Andleigh, P. K, Kiran Thakrar, “Multimedia Systems and Design”, PHI, 2003.

Reference Books

1. D. F. Rogers, J. A. Adams, “Mathematical Elements for Computer Graphics”, Second Edition, McGraw Hill International Edition, 1990.
2. F. S. Hill Jr., “Computer Graphics using OpenGL”, Second Edition, Pearson Education, 2003.
3. “The OpenGL Reference Manual - The Bluebook”, Version 1.4, Fourth Edition, Addison-Wesley.
4. Judith Jeffcoate, “Multimedia in Practice: Technology and Applications”, PHI, 1998.
5. “The OpenGL Programming Guide - The Redbook”, Version 2, Fifth Edition, Addison-Wesley.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	1	0	0	1	2	0	0	0	0	0
CO2	3	0	2	1	0	1	0	0	2	0	2	0
CO3	3	0	2	2	0	2	0	0	2	0	2	2
CO4	0	0	0	1	0	1	0	0	0	0	1	0
CO5	3	0	3	3	0	2	0	0	0	0	0	0

Course Code	:	**CSOE14**
Course Title	:	**Distributed Architecture**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**OE**

Course Objectives

- To develop an infrastructure for distributed systems architecture
- To get knowledge about universal and independent of a specific middleware technology
- To obtain knowledge about complete CORBA implementation
- To design the architecture of middleware platforms

Course Contents

UNIT I Introduction
Introduction: Distributed System - Object Model - Middleware - Sample Application - CORBA - Creation process - Application Development - The Bootstrapping Problem - Naming Service.

UNIT II ORB
ORB Architecture - Transport Layer - Presentation Layer - Interoperability Layer - Proxies - Object Services - ORB Design - ORB Functionality - Design Of MICO's ORB.

UNIT III Models
Interoperability: Model - Inter-ORB Model - Design of MICO's Interoperability Object Adapters - Terminology - Functionality - Examples of Object Adapter - Design of MICO's POA - Persistence - POA Mediator - Collocation.

UNIT IV Invocation Adapters
Functionality - Static and dynamic Invocation Interface - Design of MICO's DII-Compiler: Invocation Adapters - Compiler Fundamentals - Abstract Syntax Tree for IDL specifications - MICO's IDL Compiler.

UNIT V CORBA
CORBA Components - Web Services - Middleware for Ubiquitous Computing - case study for MICO Implementation and Application of MICO.

Course Outcomes
Upon completion of this course, the students will be able to:

- Understand the distributed system architecture
- Gain knowledge about specific middleware technology
- Implement CORBA and MICO
- Design middleware platforms
- Analyse the performance of the distributed systems by running various applications

Text Book

1. Arno Puder, Kay Römer, Frank Pilhofer, "Distributed Systems Architecture A Middleware Approach", Elsevier, 2006.

Reference Books

1. Andreas Christ, Markus, "Architectures for Distributed and Complex M-Learning Systems: Applying Intelligent Technologies", 2010.
2. Bernard I. Witt, F. Terry Baker, Everett W. Merritt, "Software Architecture and Design: Principles, Models, and Methods", 1993.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	3	2	0	3	0	0	0	0	3	0
CO2	3	0	3	0	3	2	0	0	3	0	0	0
CO3	3	2	2	0	0	3	0	0	3	0	0	0
CO4	3	3	3	0	2	0	0	0	2	0	0	3
CO5	3	2	3	0	0	2	0	0	0	3	3	3

Course Code	:	**CSOE15**
Course Title	:	**Human Computer Interaction**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	-
Course Type	:	**OE**

Course Objectives

- To provide an overview of the concepts relating to the design of human-computer interfaces
- To understand the theoretical dimensions of human factors involved in the acceptance of computer interfaces
- To understand the important aspects of implementation of human-computer interfaces
- To identify the various tools and techniques for interface analysis, design and evaluation

Course Contents

UNIT I

HCI Foundations - Input–output channels - Human memory - Thinking: reasoning and problem solving - Emotion - Individual differences - Psychology and the design of interactive systems - Text entry devices - Positioning - pointing and drawing - Display devices - Devices for virtual reality and 3D interaction - Physical controls - sensors and special devices - Paper: printing and scanning.

UNIT II

Designing - Programming Interactive systems - Models of interaction - Frameworks and HCI - Ergonomics - Interaction styles - Elements of the WIMP interface - The context of the interaction - Experience - engagement and fun - Paradigms for interaction.

Cantered Design and testing - Interaction design basics - The process of design - User focus - Scenarios - Navigation design - Screen design and layout - Iteration and prototyping.

UNIT III

HCI in the software process - Iterative design and prototyping - Design rules - Principles to support usability - Standards and Guidelines - Golden rules and heuristics - HCI patterns.

Implementation support - Elements of windowing systems - Programming the application - Using toolkits - User interface management systems.

UNIT IV

Evaluation techniques - Evaluation through expert analysis - Evaluation through user participation - Universal design - User support.

Models and Theories - Cognitive models - Goal and task hierarchies - Linguistic models - The challenge of display-based systems - Physical and device models - Cognitive architectures.

UNIT V

Collaboration and communication - Face-to-face communication - Conversation - Text-based communication - Group working - Dialog design notations - Diagrammatic notations - Textual dialog notations - Dialog semantics - Dialog analysis and design Human factors and security - Groupware - Meeting and decision support systems - Shared applications and artifacts - Frameworks for groupware - Implementing synchronous groupware - Mixed - Augmented and Virtual Reality.

Course Outcomes

Upon completion of the course, the students will be able to:

- Design and Develop processes and life cycle of Human Computer Interaction
- Analyze product usability evaluations and testing methods
- Apply the interface design standards/guidelines for cross cultural and disabled users

- Categorize, Design and Develop Human Computer Interaction in proper architectural structures
- Understand various communication approaches

Text Books

1. A Dix, Janet Finlay, G D Abowd, R Beale, "Human-Computer Interaction'', Third Edition, Pearson Publishers, 2008.
2. Shneiderman, Plaisant, Cohen, Jacobs, "Designing the User Interface: Strategies for Effective Human Computer Interaction", Fifth Edition, Pearson Publishers, 2010.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	0	0	0	1	2	0	2	1	0	1	0
CO2	2	0	0	0	0	0	0	3	0	0	2	0
CO3	0	0	2	0	2	3	0	2	2	0	2	0
CO4	1	3	2	0	0	1	0	0	0	0	2	2
CO5	0	3	0	0	2	2	0	1	0	0	0	0

Course Code	:	**CSOE16**
Course Title	:	**Image Processing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	-
Course Type	:	**OE**

Course Objectives

- To understand the fundamentals of Digital imaging
- To understand the fundamentals Image Processing techniques
- To under the mathematical transforms applicable for image processing
- To be familiar with image compression and segmentation
- Exploration of image processing algorithms

Course Contents

UNIT I Introduction
Fundamentals of Image Processing - Applications of Image Processing - Human Visual Perception - Introduction to Image Formation - Sampling and Quantization - Binary Image - Three-Dimensional Imaging - Image file formats - Color and Color Imagery: Perception of Colors.*

UNIT II Image Transformation
Fourier Transforms - Discrete Cosine Transform - Walsh-adamard Transform - Karhaunen-Loeve Transform or PCA - Discrete Wavelet Transform: Wavelet Transform - Extension to 2D Signals - Lifting Implementation of the Discrete Wave Transforms.*

UNIT III Image Enhancement and Restoration
Introduction - Distinction between image enhancement and restoration - Histogram-based Contrast Enhancement - Frequency Domain Methods of Image Enhancement - Noise Modeling - Image Restoration - Image Reconstruction.*

UNIT IV Image Segmentation and Compression
Edge detection - Edge linking via Hough transform - Thresholding - Region based segmentation - Watershed algorithm - Use of motion in segmentation - Need for data compression - Encoder-Decoder model - Types of redundancies - Lossy and Lossless compression.*

UNIT V Recognition of Image Patterns
Introduction - Decision Theoretic Pattern Classification - LDA - Bayesian Decision Theory - Texture and Shape Analysis - Case study - Image mining and Content-Based Retrieval.*

*Programming assignments are mandatory.

Course Outcomes
Upon completion of this course, the students will be able to:

- Process image enhancement and restoration techniques
- Apply image compression and segmentation Techniques
- Apply transforms for processing of images
- Design and develop image processing techniques for assisting digital forensics
- Process and recognize image patterns

Text Books

1. Maria Petrou, Costas Petrou, “Image Processing the Fundamentals”, Second Edition, John-Wiley and Sons Publishers, 2010.
2. Gonzalez, Woods, “Digital Image Processing”, Third Edition (DIP/3e), Prentice Hall, 2008.

Reference Books

1. Tinku Acharya, Ajoy K. Ray, “Image Processing Principles and Applications”, John Wiley & Sons Publishers, 2005.
2. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, “Digital Image Processing Using MATLAB”, Second Edition, Gatesmark Publishing, 2009.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	0	3	2	0	2	0	0	0	0	0	0
CO2	2	3	2	0	2	2	0	0	0	0	0	0
CO3	3	2	3	0	2	2	0	0	0	0	2	1
CO4	1	0	1	0	2	3	0	1	1	1	2	2
CO5	2	1	2	2	0	2	0	0	0	0	1	1

Course Code	:	**CSOE17**
Course Title	:	**Internet of Things**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	-
Course Type	:	**OE**

Course Objectives

- To learn the basic issues, policy and challenges in the Internet
- To get an idea of some of the application areas where Internet of Things can be applied
- To understand the cloud and internet environment
- To understand the various modes of communications with Internet
- To understand about web of things

Course Contents

UNIT I Introduction
Definition - Foundations - Challenges and Issues - Identification - Security - Components in internet of things**:** Control Units - Sensors - Communication modules - Power Sources - Communication Technologies - RFID - Bluetooth - Zigbee - Wifi - Rflinks - Mobile Internet - Wired Communication - IoT Platform Overview - Raspberry pi - Arduino boards.*

Unit II IoT Protocols
Protocol Standardization for IoT - M2M and WSN Protocols - SCADA and RFID Protocols - Issues with IoT Standardization - Protocols - IEEE 802.15.4 - BACNet Protocol - Zigbee Architecture - Network layer - APS Layer Security.*

Unit III Resource Management in the Internet of Things
Clustering - Software Agents - Data Synchronization - Clustering Principles in an Internet of Things Architecture - The Role of Context - Design Guidelines - Software Agents for Object Data Synchronization - Types of Network Architectures - Fundamental Concepts of Agility and Autonomy - Enabling Autonomy and Agility by the Internet of Things - The Evolution from the RFID-based EPC Network to an Agent based Internet of Things - Agents for the Behaviour of Objects.*

Unit IV Case Study and IoT Application Development
IoT applications in home infrastructures security Industries - IoT electronic equipments - Use of Big Data and Visualization in IoT - Industry 4.0 concepts - Sensors and sensor Node - Interfacing using Raspberry Pi/Arduino-Web Enabled Constrained Devices.*

Unit V Web of Things
Web of Things versus Internet of Things - Architecture Standardization for WoT - Platform Middleware for WoT - WoT Portals and Business Intelligence - Cloud of Things: Grid/SOA and Cloud Computing - Cloud Standards - Cloud of Things Architecture - Open Source e-Health sensor platform.*

*Programming assignments are mandatory.

Course Outcomes
Upon completion of this course, the students will be able to:

- Identify the components of IoT
- Analyze various protocols of IoT
- Design portable IoT using appropriate boards
- Develop schemes for the applications of IOT in real time scenarios
- Design business Intelligence and Information Security for WoT

Text Books

1. Honbo Zhou, “The Internet of Things in the Cloud: A Middleware Perspective”, CRC Press, 2012.
2. Dieter Uckelmann, Mark Harrison, “Architecting the Internet of Things”, Springer, 2011.

Reference Books

1. Arshdeep Bahga, Vijay Madisetti, “Internet of Things (A Hands-On-Approach)”, VPT, 2014.
2. Olivier Hersent, David Boswarthick, Omar Elloumi, “The Internet of Things – Key Applications and Protocols”, Wiley, 2012.
3. Luigi Atzori, Antonio Lera, Giacomo Morabito, “The Internet of Things: A Survey”, Journal on Networks, Elsevier Publications, October, 2010.
4. *[Online]* http://www.theinternetofthings.eu/what-is-the-internet-of-things

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	2	0	0	1	0	0	0	1	1	0
CO2	2	3	0	0	0	0	0	0	0	1	2	0
CO3	1	0	3	0	3	0	0	0	2	0	0	0
CO4	3	0	1	0	0	1	0	0	0	0	0	2
CO5	0	0	3	0	0	1	0	0	0	0	2	0

Course Code	:	**CSOE18**
Course Title	:	**Machine Learning for Engineering Applications**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**OE**

Course Objectives

- To provide a broad survey of different machine learning approaches and techniques
- To understand the principles and concepts of machine learning
- To understand neural networks concepts
- To learn regression and reinforcement learning
- To develop programming skills that helps to build real world applications based on machine learning

Course Contents

UNIT I Introduction

Introduction: Machine learning: What and why? - Types of Machine Learning - Supervised Learning - Unsupervised Learning - The Curse of dimensionality - Over and under fitting - Model selection - Error analysis and validation - Parametric vs. non-parametric models.

UNIT II Machine Learning

Types of Machine Learning - Supervised Learning - Classification models - Naïve Bayes Classifier - Decision trees - Support Vector Machines - KNN model - Dimensionality reduction - PCA.

UNIT III Clustering

Clustering approaches - Mean Shift clustering - Clustering data points and features - Bi-clustering - Multi-view clustering - K-Means clustering - K-medians clustering - Expectation Maximization (EM).

UNIT IV Neural Networks

Neural networks - Biological motivation for Neural Network - Neural network Representation - Perceptron - Feed forward networks - Multilayer Networks and Back Propagation Algorithms - Hidden layer representation - Application of neural network.

UNIT V Applications and Tools

Linear models for regression - Reinforcement Learning - Machine Learning Tools - Engineering applications.

Course Outcomes

Upon completion of this course, the students will be able to:

- Solve typical machine learning problems
- Represent data to facilitate learning
- Design and implement various machine learning algorithms for real-world applications
- Suggest supervised /unsupervised machine learning approaches for any application
- Handle tools of machine learning

Text Books

1. Kevin P. Murphy, “Machine Learning: A Probabilistic Perspective”, MIT Press, 2012.
2. Ethem Alpaydin, “Introduction to Machine Learning”, Second Edition, Prentice Hall of India, 2010.

Reference Books

1. Laurene Fausett, “Fundamentals of Neural Networks, Architectures, Algorithms and Applications”, Pearson Education, 2008.
2. Tom Mitchell, “Machine Learning”, McGraw-Hill, 1997.
3. C. M. Bishop, “Pattern Recognition and Machine Learning”, Springer, 2007.
4. Simon Haykin, “Neural Networks and Learning Machines”, Pearson 2008.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	2	0	1	0	0	0	1	0	1	0
CO2	2	0	0	3	0	0	0	0	0	0	3	0
CO3	0	0	3	0	0	3	0	0	0	0	2	3
CO4	0	3	3	0	0	2	0	0	0	1	2	0
CO5	0	3	0	0	0	3	0	0	0	0	0	2

Course Code	:	**CSOE19**
Course Title	:	**Security Principles**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**PE**

Course Objectives

- To understand the basics of Information Security
- To know the legal, ethical and professional issues in Information Security
- To know the aspects of risk management
- To become aware of various standards in this area
- To know the technological aspects of Information Security

Course Contents

UNIT I Introduction

History - What is Information Security? - Critical Characteristics of Information - NSTISSC Security Model - Components of an Information System - Securing the Components - Balancing Security and Access - The SDLC - The Security SDLC.

UNIT II Security Investigation

Need for Security - Business Needs - Threats - Attacks - Legal - Ethical and Professional Issues - An Overview of Computer Security - Access Control Matrix - Policy - Security policies - Confidentiality policies - Integrity policies and Hybrid policies.

UNIT III Security Analysis

Risk Management: Identifying and Assessing Risk - Assessing and Controlling Risk - Systems: Access Control Mechanisms - Information Flow and Confinement Problem.

UNIT IV Logical Design

Blueprint for Security - Information Security Policy - Standards and Practices - ISO 17799/BS 7799 - NIST Models - VISA International Security Model - Design of Security Architecture - Planning for Continuity.

UNIT V Physical Design

Security Technology - IDS - Scanning and Analysis Tools - Cryptography - Access Control Devices - Physical Security - Security and Personnel.

Course Outcomes

Upon completion of this course, the students will be able to:

- Discuss the basics of information security
- Illustrate the legal, ethical and professional issues in information security

- Demonstrate the aspects of risk management
- Become aware of various standards in the Information Security System
- Design and implementation of Security Techniques

Text Book

1. Michael E Whitman, Herbert J Mattord, “Principles of Information Security”, Vikas Publishing House, New Delhi, 2003.

Reference Books

1. Micki Krause, Harold F. Tipton, “Handbook of Information Security Management”, Vol. 1-3, CRC Press LLC, 2004.
2. Stuart McClure, Joel Scrambray, George Kurtz, “Hacking Exposed”, Tata McGraw Hill, 2003.
3. Matt Bishop, “Computer Security Art and Science”, Pearson/PHI, 2002.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	0	0	0	1	0	0	0	1	1	0	0
CO2	0	3	0	0	0	0	0	3	0	0	2	0
CO3	0	0	0	0	3	0	0	3	3	2	2	0
CO4	0	3	0	0	0	0	0	1	0	0	1	0
CO5	3	0	3	0	0	3	0	0	0	0	3	2

Course Code	:	**CSOE20**
Course Title	:	**Soft Computing**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	-
Course Type	:	**OE**

Course Objectives

- Describe and understand the concepts of feed forward &feedback neural networks
- Recognize the concept of fuzziness involved in various systems
- Expose the ideas about genetic algorithm
- Compare about FLC and NN toolbox
- Design algorithm for optimization problem

Course Contents

UNIT I Introduction
Introduction of soft computing - soft computing vs. hard computing - various types of soft computing techniques - applications of soft computing - Neuron-Nerve structure and synapse - Artificial Neuron and its model - activation functions - Neural network architecture - single layer and multilayer feed forward networks - McCullochPitts neuron model - perceptron model - MLP-back propagation learning methods - effect of learning rule coefficient.*

UNIT II Architecture
Counter propagation network - architecture - functioning & characteristics of counter - Propagation network - Hopfield/Recurrent network - configuration - stability constraints - associative memory - characteristics - limitations and applications - Hopfield v/s Boltzman machine - Adaptive Resonance Theory - Architecture - classifications - Implementation and training - Associative Memory.*

UNIT III Fuzzy Systems
Different faces of imprecision - inexactness - Ambiguity - Undecidability - Fuzziness and certainty - Fuzzy sets and crisp sets - Intersections of Fuzzy sets - Union of Fuzzy sets - the complement of Fuzzy sets - Fuzzy reasoning - Linguistic variables - Fuzzy propositions - Fuzzy compositional rules of inference - Methods of decompositions and defuzzification.*

UNIT IV Optimization Algorithm
Basic concept of Genetic algorithm and detail algorithmic steps - adjustment of free Parameters - Solution of typical control problems using genetic algorithm - Concept on some other search techniques like tabu search and ant colony - search techniques for solving optimization problems.*

UNIT V MATLAB Tool Box for FUZZY Logic and Neural Network
GA application to optimization problems - Case studies: Identification and control of linear and nonlinear dynamic systems using MATLAB - Neural Network toolbox - Stability analysis of Neural Network interconnection systems - Implementation of fuzzy logic controller using MATLAB fuzzy logic toolbox - Stability analysis of fuzzy control systems.*

Course Outcomes
Upon completion of this course, the students will be able to:

- Comprehend machine learning and soft computing techniques in solving real world applications
- Design and develop ML techniques with assistance of MATLAB
- Visualize and analyze behavioural pattern to develop evolutionary algorithm
- Use MATLAB toolbox
- Design Algorithm for classification Problems

Text Books

1. Timothy J. Ross, “Fuzzy Logic with Engineering Applications”, Third Edition, Wiley India, 2012.
2. Zimmermann H. J., “Fuzzy Set Theory and its Applications”, Springer International Edition, 2011.

Reference Books

1. David E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning”, Pearson Education, 2009.
2. Laurene V. Fausett, “Fundamentals of Neural Networks: Architectures, Algorithms, and Applications”, First Edition, Pearson Education, 1993.
3. W. T. Miller, R. S. Sutton, P. J. Webros, “Neural Networks for Control”, MIT Press, 1996.
4. Herniter, Marc E., “Programming in MATLAB”, Brooks/Cole-Thomson Learning, 2001.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	3	0	0	1	0	0	0	0	2	0
CO2	2	0	2	2	0	2	0	0	0	0	1	0
CO3	0	3	0	2	0	0	0	1	0	0	2	0
CO4	0	0	2	0	2	2	0	0	1	0	1	0
CO5	2	3	3	0	0	3	0	0	0	0	0	2

Course Code	:	**CSOE21**
Course Title	:	**Software Project Management**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**OE**

Course Objectives

- Recognize basic concepts and issues of software project management
- Emphasize successful software projects that support organization's strategic goals
- Comprehend software quality issues
- Comprehend software risk issues
- Analyse SPM tools

Course Contents

UNIT I SPM Concepts
Definition - components of SPM - challenges and opportunities - tools and techniques - managing human resource and technical resource - costing and pricing of projects - training and development - project management techniques.*

UNIT II Software Measurements
Monitoring & measurement of SW development - cost - size and time metrics - methods and tools for metrics - issues of metrics in multiple projects.*

UNIT III Software Quality
Quality in SW development - quality assurance - quality standards and certifications - the process and issues in obtaining certifications - the benefits and implications for the organization and its customers - change management.*

UNIT IV Risk Issues
The risk issues in SW development and implementation - identification of risks - resolving and avoiding risks - tools and methods for identifying risk management.*

UNIT V SPM Tools
Software project management using Primavera & Redmine - case study on SPM tools.*

*Programming assignments are mandatory.

Course Outcomes
Upon completion of this course, the students will be able to:

- Maintain software projects and monitor software project process
- Design and develop project modules and assign resources
- Comprehend, assess, and calculate the cost of risk involved in a project management
- Analyse the tools for risk management
- Design a Case study using SPM tools

Text Books

1. Richard H. Thayer, “Software Engineering Project Management”, Second Edition, John Wiley & Sons, 2001.
2. Royce, Walker, “Software Project Management”, Pearson Education, 2002.

Reference Books

1. Kelker S. A., “Software Project Management”, Prentice Hall, 2003.
2. Kan, Stephen H., “Metrics and Models in Software Quality Engineering”, Addison-Wesley Longman Publishing Co. Inc., 2002.

3. Galin, Daniel, “Software Quality Assurance: From Theory to Implementation”, Pearson Education India, 2004.
4. Charette, Robert N., “Software Engineering Risk Analysis and Management”, New York: McGraw Hill, 1989.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	0	0	3	0	0	2	2	3	2	0
CO2	0	0	0	0	3	0	0	2	3	3	2	0
CO3	2	0	0	0	1	0	0	2	2	2	2	0
CO4	0	0	2	0	0	2	0	0	0	0	2	0
CO5	0	0	3	2	0	3	0	2	0	0	2	0

Course Code	:	**CSOE22**
Course Title	:	**Software Testing And Practices**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**OE**

Course Objectives

- Summarize to learn the criteria for test cases
- Develop and design test cases
- Analyse test management and test automation techniques
- Assess test metrics and measurements
- Differentiate Test monitoring and controlling

Course Contents

UNIT I Introduction
Software testing - The Role process in Software Quality - Testing as a process - Overview of testing maturity model - software testing definition - Software Testing Principles - Origin of defects - Defect classes - the defect Repository and Test Design.

UNIT II Test Case Design Strategies
Testing Strategies: Testing design strategies - Test case design strategies - Black box testing - Random Testing - Equivalence partitioning - Boundary value analysis - Cause-and-Effect - State transition - Error Guessing - COTS - White box testing techniques - Statement coverage - Branch Coverage - Condition coverage - Decision/Condition coverage - Multiple condition coverage - Dataflow coverage - Mutation testing.

UNIT III Levels of Testing
The Need for Levels of Testing - unit test - Planning - Designing the unit tests - Integration test - Integration Strategies for Procedure and Functions - Integration strategies for Classes - Integration test planning - System Test: Functional Testing - Performance Testing - Stress Testing - Configuration Testing - Security Testing - Recovery Testing - Regression testing - Alpha-Beta and Acceptance Tests.

UNIT IV Object Oriented Testing
Testing Object Oriented Software: Unit Testing in OO Context - Integration Testing in OO Context - OO testing methods - Class level testing - Interclass test case design - testing for real time system

UNIT V Test Controlling and Monitoring
Controlling and Monitoring: Measurements and Milestone for Controlling and Monitoring: Status - Productivity - Cost - Error - fault and Failures - Effectiveness - Criteria for Test Completion - Reviews as testing Activity: Inspection Walkthrough - Components of review plan - testing for web application - Component level testing - Clean room tests.

Course Outcomes
Upon completion of this course, the students will be able to:

- Design test cases suitable for a software development for different domains
- Identify suitable tests to be carried out
- Prepare test planning based on the document
- Document test plans and test cases designed
- Use the automated testing tools to check the behaviour of the real time application

Text Book

1. Ilene Burnstein, “Practical Software Testing”, First Indian Reprint, Springer-Verlag, 2004.

Reference Books

1. Ali Behforooz, Frederick J Hudson, “Software Engineering Fundamentals”, Oxford University Press, New York, 2003.
2. William Perry, “Effective Methods for Software Testing”, John Wiley & Sons, Second Edition, USA, 2000.
3. Boris Beizer, “Software Testing Techniques”, Second Edition, Van Nostrand Reinhold, New York, 1990.
4. Aditya P. Mathur, “Foundations of Software Testing Fundamental Algorithms and Techniques”, Dorling Kindersley (India) Pvt. Ltd., Pearson Education, 2008.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	0	0	1	1	2	1	0	0	1	0	1	0
CO2	3	0	0	0	3	0	0	0	2	0	0	0
CO3	0	0	0	0	0	1	0	0	3	3	3	0
CO4	2	3	0	0	0	0	0	1	0	0	0	0
CO5	0	2	3	2	0	1	0	0	0	0	0	1

Course Code	:	**CSOE23**
Course Title	:	**Web Technology**
Number of Credits	:	**3-0-0-3**
Pre-requisites (Course Code)	:	**-**
Course Type	:	**OE**

Course Objectives

- Describe basics of Web Designing using HTML, DHTML, and CSS
- Categories the basics about Client side scripts and Server side scripts
- Classify web applications
- List Regular Expressions
- Create Database connectivity

Course Contents

UNIT I Web Page Designing
HTML - List - Tables - Images - Forms - Frames - Cascading Style sheets,* XML Document type definition - XML Schemas,* Document Object model.

UNIT II Scripting
Java Script - Control statements - Functions - Arrays - Objects - Events - Dynamic HTML with Java Script - Ajax.*

UNIT III Web Application
Web servers - IIS (XAMPP - LAMPP) and Tomcat Servers - Java Web Technologies - Servlets – Java Server Pages - Java Server Faces - Web Technologies in Netbeans - Building a Web Application in Netbeans - JSF Components - Session Tracking - Cookies.*

UNIT IV PHP Programming
PHP: Basics - String Processing and Regular Expressions - Form Processing and Business Logic - Using Cookies * - Dynamic Content - Operator Precedence Chart.

UNIT V JDBC
Database Connectivity with MySQL - Servlets - JSP - PHP,* Case Studies - Student information system - Health Management System.

*Programming assignments are mandatory.

Course Outcomes
Upon completion of this course, the students will be able to:

- Design and develop client side scripting techniques
- Build real world applications using client side and server side scripting languages
- Design and develop an e-governance application using web technology
- Design Database connectivity with JSP
- Design case study for student Information System and Health Management system

Text Books

1. Paul J. Deitel, Harvey M. Deitel, Abbey Deitel, “Internet & World Wide Web How to Program”, Fifth Edition, Deitel Series, 2012.
2. Jason Gilmore, “Beginning PHP and MySQL from Novice to Professional”, Fourth Edition, Apress Publications, 2010.

Reference Books

1. Robert W. Sebesta, “Programming with World Wide Web”, Fourth Edition, Pearson, 2008.
2. David William Barron, “The World of Scripting Languages”, Wiley Publications, 2000.
3. Breitman, Karin, Marco Antonio Casanova, Walt Truszkowski, “Semantic Web: Concepts, Technologies and Applications”, Springer Science & Business Media, 2007.
4. Khan, Badrul Huda et al., “Web-Based Instruction Educational Technology”, 1997.

Mapping of Course Outcomes with Programme Outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	1	1	1	0	2	0	0	0	0	2	0
CO2	2	2	1	1	0	2	0	0	3	0		0
CO3	3	3	3	0	1	3	0	0	3	3	2	0
CO4	0	0	1	1	0	1	0	0	0	0	2	0
CO5	3	3	3	0	0	3	0	0	0	0	2	2