MASTER OF SCIENCE
(COMPUTER SCIENCE)

SYLLABUS
FOR
CREDIT-BASED CURRICULUM
(Applicable from 2022-2023 onwards)

Department of Computer Applications
National Institute of Technology
Tiruchirappalli
MASTER OF SCIENCE
(COMPUTER SCIENCE)

SYLLABUS

FOR
CREDIT-BASED FLEXIBLE CURRICULUM
(APPLICABLE FROM 2022-2023 ONWARDS)

DEPARTMENT OF COMPUTER
APPLICATIONS NATIONAL INSTITUTE OF TECHNOLOGY TIRUCHIRAPPALLI-620 015,
TAMIL NADU - INDIA
### M.Sc. SYLLABUS – CORE SUBJECTS

<table>
<thead>
<tr>
<th>Semester</th>
<th>Subject Code</th>
<th>Subject Name</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>CAS 711</td>
<td>Mathematical Foundations of Computer Science</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 713</td>
<td>Networking Technologies</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 715</td>
<td>Data Structures and Algorithms</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 717</td>
<td>Problem Solving using Python and R</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 719</td>
<td>Operating Systems Fundamentals</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>CAS 701</td>
<td>Data Structures Lab</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>CAS 703</td>
<td>Python and R Lab</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>II</td>
<td>CAS 712</td>
<td>Computer Organization and Architecture</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 714</td>
<td>Theory of Computation</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 716</td>
<td>Advanced Statistical Techniques for Data Science</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 718</td>
<td>DBMS and Data Mining</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 7AX</td>
<td>Elective I</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 702</td>
<td>FOSS Lab</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>CAS 704</td>
<td>DBMS and Data Mining Lab</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>III</td>
<td>CAS 721</td>
<td>Web Computing</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 723</td>
<td>Artificial Intelligence and Machine Learning</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 725</td>
<td>Object Oriented Software Engineering</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 7BX</td>
<td>Elective II</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 7CX</td>
<td>Elective III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>CAS 705</td>
<td>Artificial Intelligence and Machine Learning Lab</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>CAS 749</td>
<td>Project Work Phase-I</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>IV</td>
<td>CAS 750</td>
<td>Project Work Phase-II</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>Grand Total</td>
<td></td>
<td>40</td>
<td>5</td>
<td>26</td>
<td>68</td>
</tr>
</tbody>
</table>

L: LECTURE | T: TUTORIAL | P: PRACTICAL | C: Credits
# LIST OF ELECTIVES

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Subject Name</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ELECTIVE – I</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAS 7A1</td>
<td>Internet of Things</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7A2</td>
<td>Cyber Security</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7A3</td>
<td>Cloud and Edge Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7A4</td>
<td>Mobile Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7A5</td>
<td>Cryptography and Blockchain Technology</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7A6</td>
<td>Computer Graphics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>ELECTIVE – II</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAS 7B1</td>
<td>Big Data Analytics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7B2</td>
<td>Computational Intelligence</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7B3</td>
<td>Computer Vision and Pattern Recognition</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7B4</td>
<td>Soft Computing Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7B5</td>
<td>Defensive and Secure Software Development</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7B6</td>
<td>Software Verification and Validation</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td><strong>ELECTIVE – III</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAS 7C1</td>
<td>GPGPU Programming</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7C2</td>
<td>Design Patterns</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CCS 7C3</td>
<td>Compiler Design</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7C4</td>
<td>Mobile Application Development</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7C5</td>
<td>Augmented Reality and Virtual Reality</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>CAS 7C6</td>
<td>Fault Tolerance Systems and Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

L: LECTURE | T: TUTORIAL | P: PRACTICAL | C: Credits
CAS 711 - MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To acquire skills in solving mathematical and logical problems.
- To comprehend mathematical principles and logic.
- To understand fundamental concepts and tools in discrete mathematics with emphasis on their applications to computer science.

Set Theory: Sets and operations, properties - power set - methods of proof - relations, graph, and matrix of a relation - partial and total orders, well ordering - equivalence relations, classes and properties - functions, 1-1, onto and bijective - composition of relations and functions - inverse functions.


Graph Theory: Definitions and basic results - Representation of a graph by a matrix and adjacency list - Trees - Cycles - Properties - Paths and connectedness - Sub graphs - Graph Isomorphism - Operations on graphs - Vertex and edge cuts - Vertex and edge connectivity, Spanning Trees, Euler circuits, Hamiltonian graphs.


Sampling Distributions and Descriptive Statistics: The Central Limit Theorem, Distributions of the sample mean and the sample variance for a normal population, Sampling distributions (Chi Square, t, F, z). Test of Hypothesis - Testing for Attributes – Mean of Normal Population – One-tailed and two-tailed tests, F-test, and Chi-Square test - Analysis of variance ANOVA – One way and two-way classifications.

REFERENCES:

COURSE OUTCOMES:

Students will be able to

- Apply the concepts of discrete mathematics in the modelling and design of computational algorithms.
CAS 713 - NETWORKING TECHNOLOGIES

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To learn state-of-the-art network reference models and computer network architectures.
- To understand the internet addressing system, design issues and functionalities of different networking layers.
- To gain comprehensive knowledge about the standard communication protocols, and recent advancement in networking technologies.


LAN Components – Packet Switching and Forwarding - LAN Technologies: Ethernet (IEEE 802.3), Token Bus, Token Ring - Wireless LAN (IEEE 802.11) - Virtual LAN (VLAN).


Point-to-Point Protocol (PPP) - Logical Link Control Protocol (LLC) - Address Resolution Protocol (ARP) - Internet Control Message Protocol (ICMP) - Dynamic Host Configuration Protocol (DHCP) - Domain Name System (DNS) – HTTP, HTTPS - File Transfer Protocol (FTP) - Simple Mail Transfer Protocol (SMTP) - Remote Administration Protocols: Telnet and Secure Shell (SSH).

REFERENCES:

COURSE OUTCOMES:
Students will be able to:

- List the functionalities of networking layers available in both OSI reference model and TCP/IP model.
- Describe available LAN and WAN Technologies.
- Describe the principles of packet switching, forwarding, and routing.
- Distinguish between TCP and UDP packet formats.
- Describe the available application protocols and networking services.
CAS 715 - DATA STRUCTURES AND ALGORITHMS

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To introduce various data structures.
- Design and analysis of algorithms and their applications.


REFERENCES:

COURSE OUTCOMES:

Students will be able to

- Develop own ADTs according to the nature of problems
- Design algorithms using various strategies
- Compute time- and space complexities of various algorithms
CAS 717 - PROBLEM SOLVING USING PYTHON AND R

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To introduce Python data structures, functions, and object-orientation concepts.
- To manipulate files using Python.
- To use Python packages for data science.
- To adopt R for statistical computing.

Python: variables, expressions, statements, operators, strings, Data structures: List, Dictionary, Tuples
Functions: parameters, return values, local and global scope, recursion, lambda functions and function composition.

Object orientation – Classes, Objects, methods, Operator overloading, and Inheritance, Method resolution order. Files: Text files, reading and writing, Handling CSV, Json files, Object serialization, Errors, and Exception handling.


R–Data sets - Data Visualization -Vector functions- Matrices and arrays- Matrix algebra- Random numbers- sampling and shuffling- Probability functions.


REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Develop solutions to real world problems using Python data structures, Functions and Object orientation concepts.
- Read and write data from/to files for processing using Python.
- Make use of Python Modules and Packages to solve complex problems.
• Perform statistical modelling using R.
CAS 719 - OPERATING SYSTEMS FUNDAMENTALS

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To understand design of an operating system and services provided by the OS.
- To understand what a process is and how processes are synchronized and scheduled.
- To acquire knowledge on different approaches to memory management.
- To understand the structure and organization of the file system and disk.
- To know the concepts of distributed and Mobile operating systems.


Disk Structure, Disk Scheduling and Disk Management. Protection and security, Distributed Operating Systems – Distributed system structure, Distributed file system, Mobile Operating systems.


REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Use system calls for managing processes, memory, and the file system.
- Explore various types of operating systems including UNIX, Linux, Android, and Windows.
- Explore the functionalities of distributed and mobile operating systems.
CAS 701 - DATA STRUCTURES LAB

Exercises for learning basic features of C/C++ and exercises to implement various data structures for real world applications.

LAB OUTCOMES:

Students will be able to:

- Write C/C++ programs for problem solving.
- Implement linear and nonlinear data structures to solve real-time problems.
- Perform searching and sorting techniques on different application domains.
- Implement different algorithm design strategies to solve complex problems.
CAS 703 - PYTHON AND R LAB

Exercises for learning basic features of Python and R Programming.

LAB OUTCOMES:

Students will be able to:

- Write Python and R programs for problem solving.
- Use Python libraries for data science.
- Use R packages for statistical modelling.
COURSE OBJECTIVES:

- To understand the data representation in a digital computer and explain how operations are performed by computer circuits.
- To understand the organization of computer, performance evaluation of memory and CPU.
- Study and analyse the modern processor architecture.


Instruction Execution – Phases of Instruction - Control Unit Operation – Control Unit Design: Hardwired, Microprogrammed, and Nano programmed Control Unit Design – Instruction Pipeline - Data Hazards - Control Hazards - Structural Hazards - Techniques for handling Hazards – Arithmetic Pipeline - Pipeline Optimization Techniques.


REFERENCES:


**COURSE OUTCOMES:**

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

- Understand basic structure and operation of digital computer.
- Understand the design of ALU to perform arithmetic and logic operations on fixed point and floating numbers.
- Understand different types of instructions and addressing modes supported in the instruction set of CPUs.
- Understand the design of control unit.
- Understand different types of memory devices, I/O communication techniques and standard I/O Interfaces.
**CAS 714 – Theory of Computation**

**COURSE PRE-REQUISITES:** Nil

**COURSE OBJECTIVES:**

- To learn the principles of what can be computed and how fast.
- To understand the applications of Computation to translators, string searching, and control circuit design.
- To learn about hierarchy of finite state machines, pushdown machines, context free grammars and Turing machines
- To learn the notions of decidability, complexity theory, recursive and recursively enumerable languages.

**FINITE AUTOMATA (FA):** Introduction, Deterministic Finite Automata (DFA) - Formal definition, simpler notations (state transition diagram, transition table), language of a DFA. Nondeterministic Finite Automata (NFA) - Definition of NFA, language of an NFA, Equivalence of Deterministic and Nondeterministic Finite Automata, Applications of Finite Automata, Finite Automata with Epsilon Transitions, Eliminating Epsilon transitions, Minimization of Deterministic Finite Automata, Finite automata with output (Moore and Mealy machines) and Inter conversion.

**REGULAR EXPRESSIONS (RE):** Introduction, Identities of Regular Expressions, Finite Automata and Regular Expressions - Converting from DFA’s to Regular Expressions, Converting Regular Expressions to Automata, applications of Regular Expressions.

**REGULAR GRAMMARS:** Definition, regular grammars and FA, FA for regular grammar, Regular grammar for FA. Proving languages to be non-regular - Pumping lemma, applications, Closure properties of regular languages.

**CONTEXT FREE GRAMMER (CFG):** Derivation Trees, Sentential Forms, Rightmost and Leftmost derivations of Strings. Ambiguity in CFG’s, Minimization of CFG’s, CNF, GNF, Pumping Lemma for CFL’s, Enumeration of Properties of CFL (Proof’s omitted).

**PUSHDOWN AUTOMATA:** Definition, Model, Acceptance of CFL, Acceptance by Final State and Acceptance by Empty stack and its Equivalence, Equivalence of CFG and PDA.

**TURING MACHINES (TM):** Formal definition and behaviour, Languages of a TM, TM as accepters and TM as a computer of integer functions, Types of TMs.

**RECURSIVE AND RECURSIVELY ENUMERABLE LANGUAGES (REL):** Properties of recursive and recursively enumerable languages, Universal Turing machine, The Halting problem, Undecidable problems about TMs. Context sensitive language and linear bounded automata (LBA), Chomsky hierarchy, Decidability, Post's correspondence problem (PCP), undecidability of PCP.

**REFERENCES:**


COURSE OUTCOMES:

Students will be able to

- Define languages by abstract, recursive definitions and regular expressions.
- Design a finite automaton to recognize a given regular language.
- Transform a language into regular expression or finite automaton or transition graph.
- Define deterministic finite automata and nondeterministic finite automata.
- Prove properties of regular languages and classify them.
CAS 716 - ADVANCED STATISTICAL TECHNIQUES FOR DATA SCIENCE

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:
- To understand advanced statistical techniques.
- To gain comprehensive knowledge on applications of statistical techniques for data analysis.


Basic multivariate statistics: multivariate descriptive statistics, multivariate distributions (normal, etc), multivariate inferential statistic.

Multivariate data- Analysis of variance (ANOVA), Multivariate analysis of variance (MANOVA)- Case study: MANOVA

Multiple linear regression- Multiple and partial correlation- Detection of Collinearity-Stepwise regression.

Validation of model assumptions- Detection of outliers- influential observation and autocorrelation

REFERENCES:

COURSE OUTCOMES:
Students will be able to:
- Apply statistical techniques for real time data analysis applications
CAS 718 - DBMS AND DATA MINING

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To learn different database models and design of databases
- To study query languages, transaction management, indexing and hashing
- To introduce the basic concepts and techniques of data mining
- To develop skills of using recent DBMS and data mining tools for solving problems.


Data warehousing – OLAP – Data Cubes - Mining Frequent Patterns, Associations and Correlations – Frequent Item set Mining Methods – Kinds of Association Rules – Constraint Based Frequent Pattern Mining.

REFERENCES:


**COURSE OUTCOMES:**

Students will be able to:

- Illustrate the features of DBMS and models for designing databases
- Apply logical database design principles in solving real world problems
- Understand the concepts, algorithms, and techniques of data mining
CAS 702 - FOSS LAB

Lab objective is to explore various open-source packages/technologies and apply them.

LAB OUTCOMES:

- To use GitHub for software development and version control in teams.
- To develop applications in FOSS environment.
- To identify and evaluate various FOSS options for any software requirement.
CAS 704 - DBMS AND DATA MINING LAB

- Exercises to construct and query databases.
- Exercises to implement data mining algorithms using ENCOG and WEKA.

LAB OUTCOMES:

Students will be able to:

- Work with ETL tools.
- Demonstrate classification, clustering in large data sets.
- Apply mining techniques for realistic data.
CAS 721 – WEB COMPUTING

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To comprehend basics of the internet and web terminologies.
- To introduce scripting language concepts for developing client-side applications.
- To practice server-side programming features – ASP .NET, PHP, JSP.
- To be familiar with database applications
- To know the usefulness of web services.

Internet and World Wide Web: Introduction to Internet, www, Internet browsers Netscape &Explorer, Introduction to Client Server Architecture/Computing, History of the web, Growth of the web, Protocols governing the web, resources of Internet, H/W & S/W requirements of Internet, Internet service providers, Internet Services, Internet Clients, and Internet Servers. Concept of E- Commerce and E-governance.


PHP: Basic command with PHP examples, Connection to server, creating database, selecting a database, listing database, listing table names creating a table, inserting data, altering tables, queries, deleting database, deleting data and tables, PHP myadmin and database bugs – Applications using PHP - Django Framework.

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Design and develop web applications.
- Understand client-side and server-side scripting and their applicability.
Introduction to Artificial Intelligence - Foundations of AI, Philosophy of artificial intelligence, problem solving, search techniques, constraint satisfaction, and game playing - minimax, Alpha beta pruning.


Representing Knowledge and reasoning in an Uncertain Domain- Bayes rule-Bayesian networks-probabilistic inference-sample applications- Planning: planning as search- partial order planning- construction and use of planning graphs.

Machine learning- Supervised learning - Regression, Classification; unsupervised learning- Clustering; Reinforcement learning, Computational Intelligence- Fuzzy systems, Swarm intelligence, neural networks models- Learning through neural nets; Basics of Deep learning

Applications of Artificial Intelligence- Natural Language Processing, Speech recognition, Computer vision, Expert systems

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Build simple knowledge-based systems.
• Apply knowledge representation and machine learning techniques to solve real-world problems.
• Apply computational Intelligence techniques.
CAS 725 - OBJECT ORIENTED SOFTWARE ENGINEERING

COURSE PRE-REQUISITES: Nil

COURSE OBJECTIVES:

- To comprehend basics of the software engineering process lifecycle.
- To introduce the object-oriented (OO) approach to software development through OO principles.
- To be conversant with UML (Unified Modelling Language) and the benefits of visual modelling/diagramming.
- To explore software engineering principles for both procedural and object-oriented approaches.


REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Practice the principles of object-oriented software development and various CASE tools.
- Convey design decisions using UML.
- Compute cost of software development and measure quality of software.
CAS 705 - ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING LAB

Exercises to implement

- Various AI and ML Techniques.

LAB OUTCOME:

Students will be able to:

- Apply the various search techniques.
- Apply knowledge representation and reasoning techniques.
CAS749 PROJECT WORK – Phase I

- To explore various research papers pertaining to chosen domain and arrive at a survey.
- To implement and demonstrate the studies done.

**Outcome:** To publish papers in conference or journal.
CAS 750 PROJECT WORK – Phase II

- Internal/External project work of 6 Months duration with submission of thesis and viva-voce examination.

**Outcome:** To publish papers in conference or journal.
CAS 7A1 – INTERNET OF THINGS

COURSE OBJECTIVES:

- To understand the fundamentals of Internet of Things (IoT).
- To study the design and working of IoT devices.
- To acquire skills to program the IoT devices.
- To develop applications using IoT Devices.


Basics of Wireless Networking, Introduction to ESP8266 Wi-Fi Module, Wi-Fi libraries, Web server- introduction, installation, configuration, Posting sensor(s) data to web server, IoT Cloud platforms, ThingSpeak API and MQTT, Interfacing ESP8266 with Web services.

IoT Case Studies and Real-World Applications: Smart Cities and Smart Homes, Connected Vehicles, Smart Grid, Smart Farming, Industrial IoT, SCADA Networks, Medical IoT, Activity Monitoring.

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- To understand the working of IoT devices.
- To program the IoT devices.
- To develop applications using IoT devices.
CAS 7A2 - CYBER SECURITY

COURSE OBJECTIVES:

- To study the concepts and principles of Cyber Security.
- To understand the security threats and the ways to mitigate them.
- To study the effectiveness of network and data security toolkits.


Cryptography: Classical Cryptography, Modern Cryptography – Forensics -watermarking - finger printing – Steganography - Biometrics


Data Security, Block chain, Bitcoins, Hacking Tools, and Techniques.

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- To understand fundamentals of cyber security.
- To study and analyze cryptographic and forensic methods.
- Analyze the network and application security issues.
- Explore the nature and logic behind security threats on the cyber space as an ethical hacker.
CAS 7A3 - CLOUD AND EDGE COMPUTING

COURSE OBJECTIVES:

- To understand Cloud Computing concepts, technologies, architecture, and applications.
- To understand different cloud programming platforms and tools to develop and deploy applications on cloud.
- To understand edge computing architecture and technologies.


Introduction to virtualization - Different Approaches to Virtualization - Server, Storage, Network Virtualization - Virtual Machine Provisioning and Manageability - VM Placement-VM Migration - Hypervisors - Case studies: VMware, KVM, Xen – Containers

Service Science - Service Oriented Architecture - Web Services: SOAP, WSDL, UDDI - Web Services Discovery and Composition - REST based Web Services

Cloud Computing Architecture - Cloud Computing Service Delivery Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS) Deployment Models: Public cloud, Private cloud, Hybrid cloud - Data Centre Design and Management - Service Level Agreements (SLAs) – Pricing Models of Cloud - Migrating to Cloud – Cloud Simulators - Cloud Security Risks - Case Studies: Amazon AWS, Microsoft Azure, Amazon EC2, Google Cloud

Edge computing Concepts - Reference Architecture - Edge Computing in the IoT–Edge Analytics and Data Management -Use Cases

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Acquire Knowledge on the concepts and technologies of Cloud Computing.
• Define the principles of virtualization
• Identify the Service Oriented Architecture for Distributed Computing workflow.
• Acquire Knowledge on the concepts and technologies of Edge Computing.
COURSE OBJECTIVES:

- To be familiar with mobile technologies, protocols, architectures, and applications.
- To gain comprehensive knowledge about the state-of-the-art mobile networks.

Detailed Introduction of Mobile Computing: History, Types, Benefits, Application, Evolution, Security Concern regarding Mobile Computing, Different Propagation Modes, Wireless Architecture, and its types, needs of mobile user, The cellular concept: Cellular system, hexagonal geometry cell and concept of frequency reuse, Channel Assignment Strategies Distance to frequency reuse ratio

Telecommunication System: GSM: - Channel allocation, call routing Architecture, PLMN interface, addresses and identifiers, network aspects, frequency allocation, authentication and security, Handoffs Technique. GPRS: network operation, data services, Applications, Billing and charging

Mobile IP: Need of mobile IP, IP packet delivery, Agent Discovery, Registration, Tunnelling and encapsulation, Route optimization, IP Handoff


REFERENCES:


COURSE OUTCOMES:

Students will be able to:
• Understand concepts of Mobile Communication.
• Analyse next generation Mobile Communication System.
• Understand network and transport layers of Mobile Communication.
• Analyze various protocols of all layers for mobile and ad hoc wireless communication networks.
• Understand IP and TCP layers of Mobile Communication.
**COURSE OBJECTIVES:**

- To understand the basic network security concepts.
- To acquire knowledge of cryptographic algorithms.
- To understand the history, types, and applications of Blockchain.
- To acquire knowledge about cryptography and consensus algorithms.


Cryptographic algorithms - Number Theory- Modern Block Ciphers: DES, 3DES, AES, Blowfish, CAST-128 - Stream Cipher - Public Key Cryptography: RSA, Diffie-Hellman, Elgamal, ECC.


Introduction to Blockchain - Distributed DBMS – Limitations of Distributed DBMS, Introduction to Block chain – History, Definition, Distributed Ledger, Blockchain Categories – Public, Private, Consortium, Blockchain Network and Nodes, Peer-to-Peer Network, Mining Mechanism, Generic elements of Blockchain, Features of Blockchain, and Types of Blockchain.

Blockchain Architecture - Operation of Bitcoin Blockchain, Blockchain Architecture – Block, Hash, Distributer P2P, Structure of Blockchain- Consensus mechanism: Proof of Work (PoW), Proof of Stake (PoS), Byzantine Fault Tolerance (BFT), Proof of Authority (PoA) and Proof of Elapsed Time (PoET)

**REFERENCES:**


**COURSE OUTCOMES:**

Students will be able to:

- Recall the basic network security concepts.
- Apply Cryptographic Algorithms.
- Describe the history, types, and applications of Blockchain.
- Gain familiarity with cryptography and Consensus algorithms.
CAS 7A6 - COMPUTER GRAPHICS

COURSE OBJECTIVES:

- To provide a comprehensive introduction to computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends.
- To understand computer graphics techniques (2-D/3-D), focusing on 3D modelling, image synthesis, and rendering.
- Introduce geometric transformations, geometric algorithms, software systems (OpenGL), 3D object models (surface, volume and implicit), visible surface algorithms, image synthesis, shading and mapping, ray tracing, radiosity, global illumination, photon mapping, and anti-aliasing.
- To explore the interdisciplinary nature of computer graphics which is emphasized in the wide variety of examples and applications.

Types of computer graphics, Graphic Displays- Random scan displays, Raster scan displays, Frame buffer and video controller, Points and lines, Line drawing algorithms, Circle generating algorithms, Midpoint circle generating algorithm, and parallel version of these algorithms.

Basic transformation, Matrix representations and homogenous coordinates, Composite transformations, Reflections, and shearing. Windowing and Clipping: Viewing pipeline, viewing transformations, 2-D Clipping Algorithms- Line clipping algorithms such as Cohen Sutherland line clipping algorithm, Liang Barsky algorithm, Line clipping against nonrectangular clip windows; Polygon clipping – Sutherland Hodgeman polygon clipping, Weiler and Atherton polygon clipping, Curve clipping, Text clipping.

Compression - Text and Image Compression: Introduction, compression principles, text compression, image compression, Audio and Video Compression: Introduction, audio compression, DPCM, ADPCM, APC, LPC, video compression, video compression principles, H.261, H.263, MPEG, MPEG-1, MPEG-2, and MPEG-4

3-D geometric primitives, 3-D Object representation, 3-D Transformation, 3-D viewing, projections, 3-D Clipping, Quadric surfaces, Spheres, Ellipsoid, Blobby objects, introductory concepts of Spline, Bspline and Bezier curves and surfaces.


REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Understand concepts of computer graphics.
- Understand different computer graphic techniques.
CAS 7B1 - BIG DATA ANALYTICS

COURSE OBJECTIVES:

- To introduce big data analytics and to understand the importance of big data.
- To introduce different approaches of exploiting big data sources such as social media, mobile devices, and sensors.
- To understand methodologies of analyzing big data.
- To acquire knowledge of handling unstructured and semi-structured data using NoSQL database.


HADOOP – Hadoop Ecosystem, Hadoop Distributed File System, Components of Hadoop, Analysing the Data with Hadoop, Scaling Out, Hadoop Streaming, HDFS Design; MapReduce – MapReduce Framework, Techniques to Optimize MapReduce Jobs, Anatomy of a Map Reduce Job Run, Failures, Job Scheduling, Shuffle and Sort, Task execution, Developing MapReduce Application.

NoSQL DATABASES – NoSQL, Characteristics of NoSQL, History of NoSQL, Types of NoSQL Data Models, CAP Theorem, MongoDB, Neo4j.

FRAMEWORKS – Hive Services, Data Types in Hive, Built-in Functions in Hive, Hive QL; The Pig architecture, Benefits and Properties of Pig, Pig Latin, Pig Operators; HBase, ZooKeeper, Mahout.

DATA VISUALIZATION – Representing Visual Data, Techniques for Visual Data Representation, Types of Data Visualization, Applications of Data Visualization, Visualizing Big Data, Data Visualization Tools, Data Visualization with Tableau.

REFERENCES:

COURSE OUTCOMES:

Students will be able to:

- Comprehend the concepts of big data analytics.
- Build web-intelligence applications exploiting big data using big data platforms based on the ‘map-reduce’ parallel programming framework.
- Effectively use NoSQL database for storage and retrieval of big data.
Introduction to Computational Intelligence - Intelligence machines - Computational Intelligence paradigms – Fuzzy logic - Fuzzy relationships - Fuzzy Sets - Operations on Fuzzy sets - Fuzzy rules - Fuzzy inference systems - Fuzzy expert systems - Applications of Fuzzy Set theory to different branches of science and engineering


Evolutionary computation – Chromosomes - Fitness functions - Selection mechanisms - Genetic algorithms - Crossover - mutation – Convergence – Applications - Genetic programming - Evolution strategies - Evolutionary neural network - Case studies


REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Apply intelligent technologies for various applications.
- Model global optimization solutions for various real-life problems.
CAS 7B3 - COMPUTER VISION AND PATTERN RECOGNITION

**COURSE OBJECTIVES:**

- To understand the fundamental concepts related to image processing, feature extraction, and pattern analysis.
- To apply the concepts to solve computer vision problems of different fields.

Fundamentals of Image Formation, Transformation: Orthogonal, Euclidean, Affine, Projective, etc; Fourier Transform, Convolution and Filtering, Image Enhancement-Histogram Processin

Edges - Canny, LOG, DOG; Line detectors (Hough Transform), Corners - Harris and Hessian Affine, Orientation Histogram, SIFT, SURF, HOG, GLOH, Scale-Space Analysis- Image Pyramids and Gaussian derivative filters, Gabor Filters and DWT.

Image Segmentation: Region Growing, Edge Based approaches to segmentation, Graph-Cut, Mean-Shift, MRFs, Texture Segmentation; Object detection.

Motion analysis: Background Subtraction and Modeling, Optical Flow, KLT, Spatio-Temporal Analysis, Dynamic Stereo; Motion parameter estimation.


**REFERENCES:**


**COURSE OUTCOMES:**

Students will be able to:

- Apply fundamental algorithms in Image Processing and analyze their applicability for real time problems.
- Design solutions for various computer vision and pattern recognition problems.
COURSE OBJECTIVES:

- To learn the techniques of soft computing.
- To learn the applications of evolutionary and genetic algorithms.
- To know the design of fuzzy controller and rough sets.
- To know the hybridization of soft computing systems.


Artificial Neural Network (ANN): Biological foundation of Neural Network, Neural Model and Network Architectures, Perceptron Learning, Supervised Hebbian Learning, Backpropagation, Associative Learning, Competitive Networks, Hopfield Network, Computing with Neural Nets, and applications of Artificial Neural Network


REFERENCES:


COURSE OUTCOMES:

Students will be able to:

• Explain the basics of soft computing.
• Apply neural network principles and algorithms for given problems.
• Apply the principles of fuzzy and hybrid algorithms for real time applications.
• Solve problems using evolutionary algorithms.
Introduction - Types of security; Correctness vs. security; Assets and threats; Security mindset; Layers of security – Software security - Why it is important; Security in the Software Development Lifecycle (SDLC); Kinds of attacks; How to design security in; Defense in depth and breadth; Principles, tactics, guidelines, and standards.

Programming Language Overviews - Overviews of secure software of JavaScript, Java, C, and C++ - Security in design - Domain-driven design; Requirements definitions; Threat models - Security in Programming - Defensive coding; Immutability: Design by contract; Validation (origin, size, bounds, lexical, syntactic, and semantic checking); Integrity maintenance (defensive copying, failing fast, managing exceptions); Secure error handling; Mitigating data leaks; Reducing complexity.

Security in testing and operations - Types of testing (including fuzz testing and pen testing); SecOps; Continuous integration – Language dependent guidelines and standards for secure programming in C, C++, Java, and JavaScript – Cryptology - math and the algorithms behind cryptography, cryptanalysis, encryption, and hashing

Authentication and Authorization - difference between Authentication and Authorization; Passwords, hashing and salting; tokens; Defining a good permission system; Digital signatures; SSO; SAML and OAuth.

Web security - OWASP - CWE and CVE - compare Firebase with systems that use a traditional backend - Network Security: Internet architecture and security issues with TCP/IP; what is transport-layer security and how does it work; AWS IAM – Privacy - Privacy vs. security; Expectations

REFERENCES:

COURSE OUTCOMES:

Students will be able to:

- Understand software security fundamentals.
- Develop secure software.
- Use secure coding practices in software development.
COURSE OBJECTIVES:

- To understand the proof methods for software verification,
- To write human-readable, machine-checkable proofs of software correctness,
- To design and develop reusable formal proofs, e.g., of safety/liveness for concurrent programs.


for Testing, Metrics for Maintenance, Software rejuvenation, Reusability and Portability, Estimation of maintenance efforts, Configuration management, Regression testing, Emerging Technologies.

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Understand the basic concepts of software engineering.
- Practice the application principles of Software testing.
- Analyze the process of software verification and validation.
CAS 7C1 - GPGPU PROGRAMMING

COURSE OBJECTIVES:

- To introduce the features of massively parallel programming architecture.
- To utilize massively parallel computing capability of a GPU for high performance computing requirements.
- To provide an overview of parallel design paradigms.


Parallel Programming Paradigms: Overview, Element Addressing - Multidimensional Kernel, Map, Gather, Scatter, Reduce, Scan, Thread Handling, Overview, Barrier Synchronization, Thread Synchronization Demo, Warp Divergence, Matrix Multiplication

CUDA Tools and APIs: Tools Overview, Using NSight Visual Studio and Eclipse, Running CUDA Apps, Debugging, Profiling, CUDA Architecture, CUDA APIs, CUDA 5.5 and 6 Features

CUDA programming: Overview, Compilation Process, Von Neumann Processor and CUDA Thread, Execution Model, first program in CUDA (Vector Addition), Location Qualifiers, Grid and Block Dimensions, Global Memory, Constant and Texture Memory, Shared Memory, Register and Local Memory, Data Movement, Error Handling, Device Introspection

Atomics: Overview, Need for Atomics, Atomic Functions, Atomic Sum, Monte Carlo Pi, Handling Events and Streams, Overview, Events, Event API, Event example, Pinned.

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Analyze GPU Design for finding solutions.
- Understand parallel programming paradigms.
- Solving high performance computing problems using GPUs.
COURSE OBJECTIVES:

- To comprehend the rationale and benefits of software design patterns.
- To impart knowledge on the development of good design patterns.

Introduction: Introduction to Design Patterns, Object Oriented Analysis and Design, Types of Design Patterns, Applications of Design Patterns, Anti Patterns, Code Refactoring Techniques for design patterns

Creational Patterns: Factory Methods, Static Factory Pattern, Singleton Pattern, Abstract Factory Pattern, Object Pool Pattern, Prototype Pattern, Builder Pattern, Telescopic Constructor Pattern

Structural Patterns: Adapter Pattern, Bridge Pattern, Composite Pattern, Decorator Pattern, Façade Pattern, Flyweight Pattern, Private Class Data, Proxy Pattern

Behavioral Design Patterns - I: Chain of responsibility Pattern, Command Pattern, Interpreter Pattern, Iterator Pattern, Mediator Pattern

Behavioral Design Patterns – II: Memento Pattern, Null Object Pattern, Observer Pattern, State Pattern, Strategy Pattern, Template method, Visitor Pattern

REFERENCES:

2. Robson, Elisabeth., and Freeman, Eric, “Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software”, O'Reilly, 2021.

COURSE OUTCOMES:

Students will be able to:

- Solve common problems in software design with ease.
- Analyze object-oriented design for patterns.
- Represent design decisions more effectively with examples and architectural use cases.


REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Explain various phases of compiler.
- Explain various parsing techniques.
- Explain the process of intermediate code generation.
- Explain various code optimization techniques.
- Apply the knowledge of lex tool and yacc tool to develop a scanner and parser, respectively.
Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements gathering and validation for mobile applications

Basic Design - Introduction – Basics of embedded systems design – Embedded OS - Design constraints for mobile applications, both hardware and software related – Architecting mobile applications – user interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability, and modifiability.

Advanced Design - Designing applications with multimedia and web access capabilities – Integration with GPS and social media networking applications – Accessing applications hosted in a cloud computing environment – Design patterns for mobile applications


iOS - Introduction to Objective C – iOS features – UI implementation – Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using WiFi - iPhone marketplace.

REFERENCES:


COURSE OUTCOMES:

Students will be able to:

- Describe the requirements for mobile applications.
- Develop design for mobile applications for specific requirements.
- Implement the design using mobile application development frameworks.


Development Tools and Frameworks in Virtual Reality: Frameworks of Software Development Tools in VR. X3D Standard; Vega, MultiGen, Virtools etc, 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.

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.

REFERENCES:

COURSE OUTCOMES:

Students will be able to:

- Understand the basic concepts of Augmented Reality and Virtual Reality (AR and VR).
CAS 7C6 FAULT TOLERANCE SYSTEMS AND TECHNIQUES

COURSE OBJECTIVES:
- To understand the principles of fault-tolerant systems and techniques.
- To understand the reliability and security of software in the sense of fault tolerance.

Introduction to Fault-Tolerance: Error, Faults and Failures; Reliability and Availability; Dependability Measures, Mathematical Reliability Modeling: Probability Basics; Reliability and Availability Modeling, Analysis using Markov Models

Hardware Fault-Tolerance: Canonical and Resilient Structures; Reliability Evaluation Techniques and Models; Processor-level Fault Tolerance; Byzantine Failures and Agreements

Information Redundancy: Error Detection/Correction Codes (Hamming, Parity, Checksum, Berger, Cyclic, Arithmetic); Encoding/Decoding circuits; Resilient Disk Systems (RAID), Fault-Tolerant Networks: Network Topologies and their Resilience; Fault-tolerant Routing

Software Fault-Tolerance: Single-Version Fault Tolerance; N-Version Programming; Recovery Approach; Exception and Conditional (Assert) Handling; Reliability Models, Checkpointing: Optimal Checkpointing; Checkpointing in Distributed and Shared-memory Systems

Fault-Tolerant System Design/Applications: Defect-tolerance in VLSI Designs; Fault Detection in Cryptographic Systems

REFERENCES:
2. Elena Dubrova; Fault-Tolerant Design; Springer, 2013.

COURSE OUTCOMES:
Students will be able to:
- Understand the basic concepts of fault tolerance.
- Practice the application principles of fault tolerance.
- Analyze the fault tolerance in hardware and software level.