

ENTREPRENEURSHIP DEVELOPMENT (Common for CSE, DBS, SE)

L	T	P	C
1	0	1	1

Er.-Entrepreneurship-Enterprise: Conceptual issues - Entrepreneurship vs. Management - Roles and functions of Er in relation to the enterprise and in relation to the economy - Entrepreneurship is an interactive process between the individual and the environment - Small business as seedbed of Entrepreneurship - Entrepreneur competencies - Entrepreneur motivation - performance and rewards.

- The teachers should emphasize to students the desirability as well as feasibility of a career in Entrepreneurship in the Indian scenario.
- The teachers may make use of Entrepreneurship Development Institute of India's Inventory of Entrepreneur Competencies and National Institute of Entrepreneurship, Developments training kit for arousing Entrepreneur motivation and capacity and capability building.

Opportunity scouting and idea generation: role of creativity and innovation and business research - Sources of business ideas - Entrepreneur opportunities in contemporary business environment, for example opportunities in net-work marketing, franchising, business process outsourcing in the early 21 century.

- The students may be advised to visit various product/service franchises, BPO concerns and meet up/down links in the net-work marketing.

Launching of Micro, Small and Medium Enterprises (MSMEs): Decision to be self employed-Selection of product and process technology – Location of Enterprise- Forms of Business organization- Preparation of Project Report- Registration/ Statutory Licenses-Clearances

- The students may be advised to develop a structured instrument [questionnaire] for conducting surveys of the various aspects of entrepreneur/enterprise. They may also be advised to prepare a comprehensive business plan. The desirability and feasibility of liaison with relevant funding and non-funding agencies may also be explored.

REFERENCES:

1. Brandt, Steven C., The 10 Commandments for Building a Growth Company, Third Edition, Macmillan Business Books, Delhi, 1977.
2. Bhide, Amar V., The Origin and Evolution of New Business, Oxford University Press, New York, 2000.
3. Dollinger M.J., 'Entrepreneurship strategies and Resources', 3rd edition, Pearson Education, New Delhi 2006.
4. Desai, Vasant Dr. (2004) Management of small scale enterprises New Delhi: Himalaya Publishing House,
5. Taneja, Gupta, Entrepreneur Development New Venture Creation, 2nd ed. Galgotia Publishing Company
6. Holt, David H., Entrepreneurship: Strategies and Resources, Illinois, Irwin, 1955.

7. Panda, Shiba Charan, Entrepreneurship Development, New Delhi, Anmol Publications.
8. Patel, V.G., The Seven Business Crises and How to Beat Them, Tata-Mcgraw, New Delhi, 1995.
9. SIDBI Report n Small Scale Industries Sector[latest edition]
10. Verma, J.C., and Gurpal Singh, Small Business and Industry-A Handbook for Entrepreneurs, Sage, New Delhi, 2002
11. Vesper, Karl H., New Venture Strategies, [Revised Edition], New Jersy, Prentice Hall, 1990.
12. Hisrich, Entrepreneurship, Tata McGraw Hill, New Delhi, 2001.
13. S.S.Khanka, Entrepreneurial Development, S.Chand and Company Limited, New Delhi, 2001.
14. Mathew Manimala, Entrepreneurship Theory at the Crossroads, Paradigms & Praxis, Biztrantra , 2ndEdition , 2005.
15. Prasanna Chandra, Projects –Planning, Analysis, Selection, Implementation and Reviews, Tata McGraw-Hill, 1996.
16. P.Saravanavel, Entrepreneurial Development, Ess Pee kay Publishing House, Chennai - 1997.
17. Arya Kumar, Entrepreneurship, Pearson. 2012.
18. Donald F Kuratko, T.V Rao, Entrepreneurship: A South Asian perspective, Cengage Learning, 2012.

IMAGE PROCESSING AND PATTERN RECOGNITION
(Common to CSE, DBS, SE)

L	T	P	C
3	0	0	3

Course objectives:

To introduce the student to various Image processing and Pattern recognition techniques.

- To study the Image fundamentals.
- To study the mathematical morphology necessary for Image processing and Image segmentation.
- To study the Image Representation and description and feature extraction.
- To study the principles of Pattern Recognition.
- To know the various applications of Image processing.

INTRODUCTION

Elements of an Image Processing System- Mathematical Preliminaries- Image Enhancement- Grayscale Transformation- Piecewise Linear Transformation- Bit Plane Slicing- Histogram Equalization--Histogram Specification- Enhancement by Arithmetic Operations- Smoothing Filter- Sharpening Filter- Image Blur Types and Quality Measures.

MATHEMATICAL MORPHOLOGY AND IMAGE SEGMENTATION

Binary Morphology-Opening and Closing- Hit-or-Miss Transform- Grayscale Morphology- Basic morphological Algorithms- Morphological Filters-Thresholding-Object (Component) Labeling-Locating Object Contours by the Snake Model- Edge Operators-Edge Linking by Adaptive Mathematical morphology- Automatic Seeded Region Growing- A Top-Down Region Dividing Approach.

IMAGE REPRESENTATION, DESCRIPTION AND FEATURE EXTRACTION.

Run-Length Coding- Binary Tree and Quadtree- Contour Representation-Skeletonization by Thinning- Medial Axis Transformation-Object Representation and Tolerance- Fourier Descriptor and Moment Invariants-Shape Number and Hierarchical Features-Corner Detection- Hough Transform-Principal Component Analysis-Linear Discriminate Analysis- Feature Reduction in Input and Feature Spaces

PATTERN RECOGNITION

The Unsupervised Clustering Algorithm-Bayes Classifier- Support Vector Machine- Neural Networks-The Adaptive Resonance Theory Network-Fuzzy Sets in Image Analysis-Document image processing and classification-Block Segmentation and Classification- Rule-Based Character Recognition system- Logo Identification-Fuzzy Typographical Analysis for Character Pre classification-Fuzzy Model for Character Classification.

APPLICATIONS:

Face and Facial Feature Extraction-Extraction of Head and Face Boundaries and Facial Features- Recognizing Facial Action Units-Facial Expression Recognition in JAFFE Database-Image Steganography- Types of Steganography- Applications of Steganography- Embedding Security and Imperceptibility- Examples of Steganography Software-Genetic Algorithm Based Steganography.

REFERENCES:

8. Image Processing and Pattern Recognition: Fundamentals and Techniques- Frank Y Shih, Willey IEEE Press, April 2010.
9. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins,|| Digital Image Processing using MATLAB||, Pearson Education, Inc., 2004.
10. D.E. Dudgeon and R.M. Mersereau, -Multidimensional Digital Signal Processing||,
11. Prentice Hall Professional Technical Reference, 1990.
12. William K. Pratt, -Digital Image Processing||, John Wiley, New York, 2002.
13. Milan Sonka et al, -Image Processing, Analysis and Machine Vision||, Brookes/Cole, Vikas Publishing House, 2nd edition, 1999;
14. Sid Ahmed, M.A., - Image Processing Theory, Algorithms and Architectures||, McGrawHill, 1995.

L	T	P	C
3	0	2	4

DATA WAREHOUSING AND MINING (Common to CSE & DBS)

This course will provide the core concepts of data mining, its techniques, implementation, benefits, and outcome expectations from this new technology. It will also identify industry branches which most benefit from Data warehousing and data mining. The course will also focus on business solutions and the advanced applications of Data mining.

INTRODUCTION: Fundamentals of data mining - Data Mining Functionalities - Classification of Data Mining systems - Data Mining Task Primitives, Integration of a Data Mining System with a Database or a Data Warehouse System, Major issues in Data Mining. Data Preprocessing: Need for Preprocessing the Data, Data Cleaning, Data Integration and Transformation, Data Reduction, Discretization and Concept Hierarchy Generation.

DATA WAREHOUSE AND OLAP TECHNOLOGY FOR DATA MINING: Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, Further Development of Data Cube Technology, From Data Warehousing to Data Mining Data Cube Computation and Data Generalization: Efficient Methods for Data Cube Computation, Further Development of Data Cube and OLAP Technology, Attribute-Oriented Induction.

MINING FREQUENT PATTERNS, ASSOCIATIONS AND CORRELATIONS: Basic Concepts, Efficient and Scalable Frequent Item set Mining Methods, Mining various kinds of Association Rules, From Association Mining to Correlation Analysis, Constraint-Based Association Mining, Classification and Prediction: Issues Regarding Classification and Prediction, Classification by Decision Tree Induction, Bayesian Classification, Rule-Based Classification, Classification by Back propagation, Support Vector Machines, Associative Classification, Lazy Learners, Other Classification Methods, Prediction, Accuracy and Error measures, Evaluating the accuracy of a Classifier or a Predictor, Ensemble Methods

CLUSTER ANALYSIS INTRODUCTION :Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, Model-Based Clustering Methods, Clustering High-Dimensional Data, Constraint-Based Cluster Analysis, Outlier Analysis.

ADVANCED DATA MINING: Web mining – Text Mining – Spatial and Temporal Mining - Applications and Trends in Data Mining: Data Mining Applications, Data Mining System Products and Research Prototypes, Additional Themes on Data Mining and Social Impacts of Data Mining – Case studies.

LAB COMPONENTS (Only for DBS)

The students are expected to implement assigned lab exercises.

1. For a given set of data, perform and implement the appropriate pre-processing.
2. Implement the Apriori and Frequent Pattern Tree mining algorithm to identify the interesting pattern from the given data set.

3. Implement (i) Bayesian (ii) K-nearest Neighbour (iii) Decision tree based method and analyze the result using any one of the performance measure.
4. Implement (i) K-means (ii) Hierarchical algorithm for a given data set
5. Identify the application and Implement using the suitable techniques.

REFERENCES

1. Jiawei Han and Micheline Kamber , -Data Mining – Concepts and Techniquesl, Morgan Kaufmann Publishers, Elsevier, 2nd Edition, 2006.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, -Introduction to Data Miningll, Pearson education.
3. Arun K Pujari, -Data Mining Techniquesl, 2nd edition, Universities Press.
4. Sam Aanhory & Dennis Murray, -Data Warehousing in the Real Worldl, Pearson Edn Asia.
5. K.P.Soman, S.Diwakar,V.Ajay , -Insight into Data Miningll, PHI, 2008.
6. Paulraj Ponnaiah , -Data Warehousing Fundamentalsl, Wiley student Edition

BIG DATA ANALYTICS (Common to CSE & DBS)

Course Objectives:

- To understand big data analytics as the next wave for businesses looking for competitive advantage
- To understand the financial value of big data analytics
- To explore tools and practices for working with big data
- To understand how big data analytics can leverage into a key component
- To understand how to mine the data
- To learn about stream computing
- To know about the research that requires the integration of large amounts of data

INTRODUCTION TO BIG DATA

Analytics – Nuances of big data – Value – Issues – Case for Big data – Big data options Team challenge – Big data sources – Acquisition – Nuts and Bolts of Big data. Features of Big Data - Security, Compliance, auditing and protection - Evolution of Big data – Best Practices for Big data Analytics - Big data characteristics - Volume, Veracity, Velocity, Variety – Data Appliance and Integration tools – Greenplum – Informatica

DATA ANALYSIS

Evolution of analytic scalability – Convergence – parallel processing systems – Cloud computing – grid computing – map reduce – enterprise analytic sand box – analytic data sets – Analytic methods – analytic tools – Cognos – Microstrategy - Pentaho. Analysis approaches – Statistical significance – business approaches – Analytic innovation – Traditional approaches – Iterative

STREAM COMPUTING

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 - Realtime Analytics Platform(RTAP) applications IBM Infosphere – Big data at rest – Infosphere streams – Data stage – Statistical analysis – Intelligent scheduler – Infosphere Streams

PREDICTIVE ANALYTICS AND VISUALIZATION

Predictive Analytics – Supervised – Unsupervised learning – Neural networks – Kohonen models – Normal – Deviations from normal patterns – Normal behaviours – Expert options – Variable entry - Mining Frequent itemsets - Market based model – Apriori Algorithm – Handling large data sets in Main memory – Limited Pass algorithm – Counting frequent itemsets in a stream – Clustering Techniques – Hierarchical – K- Means – Clustering high dimensional data Visualizations - Visual data analysis techniques, interaction techniques; Systems and applications

FRAMEWORKS AND APPLICATIONS

IBM for Big Data – Map Reduce Framework - Hadoop – Hive - – Sharding – NoSQL Databases - S3 - Hadoop Distributed file systems – Hbase – Impala – Analyzing big data with twitter – Big data for E-Commerce – Big data for blogs.

LAB COMPONENTS (Only for DBS)

The students are expected to implement assigned lab exercises.

- Work with big data platform and its analysis techniques.
- Analyze the big data for useful business applications.
- Select visualization techniques and tools to analyze big data
- Implement search methods and visualization techniques
- Design efficient algorithms for mining the data from large volumes.
- Explore the technologies associated with big data analytics such as NoSQL, Hadoop and MapReduce

REFERENCES:

11. Frank J Ohlhorst, -Big Data Analytics: Turning Big Data into Big Money, Wiley and SAS Business Series, 2012.
12. Colleen Mccue, -Data Mining and Predictive Analysis: Intelligence Gathering and Crime Analysis, Elsevier, 2007
13. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
14. Anand Rajaraman and Jeffrey David Ullman, Mining of Massive Datasets, Cambridge University Press, 2012.
15. Bill Franks, -Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, Wiley and SAS Business Series, 2012.
16. [Paul Zikopoulos](#), [Chris Eaton](#), [Paul Zikopoulos](#), -Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, McGraw Hill, 2011.
17. [Paul Zikopoulos](#), [Dirk deRoos](#), [Krishnan Parasuraman](#), [Thomas Deutsch](#) , [James Giles](#), [David Corrigan](#), -Harness the Power of Big data – The big data platform, McGraw Hill, 2012.
18. Glenn J. Myatt, Making Sense of Data, John Wiley & Sons, 2007
19. Pete Warden, Big Data Glossary, O'Reilly, 2011.
20. Jiawei Han, Micheline Kamber -Data Mining Concepts and Techniques, Second Edition, Elsevier, Reprinted 2008.

L	T	P	C
3	0	2	4

ADVANCED OPERATING SYSTEMS (Only for CSE)

Course objectives: To study the advanced concepts of Operating Systems, Distributed Systems, multiprocessor, real-time, database and mobile operating systems.

INTRODUCTION: Functions of operating systems, Design approaches: layered, kernel based and virtual machine approach, why advanced operating systems, types of advanced operating systems.

DISTRIBUTED OPERATING SYSTEMS: Architecture of distributed operating systems, system architecture types, issues in distributed operating systems, inherent limitation of distribute systems, distributed mutual exclusion: classification of mutual exclusion algorithms, Lamport's token based algorithm, Suzuki-Kasami's Broadcast algorithm, Raymond's Tree based algorithm, Distributed deadlock detection, Distributed file systems, Distributed shared memory, Distributed scheduling.

MULTIPROCESSOR OPERATING SYSTEMS: Introduction, structure of multiprocessor operating system, operating system design issues, threads, the test and set instruction, the swap instruction, implementation of the process wait , processor scheduling, reliability and fault tolerance.

REAL TIME OPERATING SYSTEM: Introduction to Real time systems and Real Time Operating Systems, Characteristics of Real Time operating Systems, Classification of Real Time Operating Systems, Services, structure, goal and feature of RTOS, architecture of RTOS, micro kernels and monolithic kernels, tasks in RTOS, Performance measures, estimating program runtimes, task assignment, scheduling in RTOS, rate monotonic scheduling, priority inversion, task management, inter task communication, applications of various RTOS.

DATA BASE OPERATING SYSTEMS: Introduction to database operating systems, concurrency control: theoretical aspect, distributed database system, concurrency control algorithms.

MOBILE OPERATING SYSTEM: Symbian O.S.: introduction, kernel design in Symbian OS, scheduling in Symbian OS, File systems on mobile phones, I/O in Symbian OS, Application development using Android, Introduction to cloud OS.

LAB COMPONENTS: (Only for CSE)

The students are expected to implement assigned lab exercises. They are also expected to carry out one or two case studies.

1. Implementation of Process scheduling algorithms.
2. Simulation of Deadlock detection, prevention and recovery process.
3. Implementation of Distributed mutual exclusion Algorithms.
4. Implementation of Distributed OS Agreement protocols.

5. Implementation of Distributed OS Resource Scheduling algorithms
6. Two-Phase Commit Protocol in Distributed OS.
7. CASE STUDY: Linux on ARM(OS for embedded systems)
8. CASE STUDY: Development of a reasonably sized dynamically loadable kernel module for Linux kernel
9. CASE STUDY: Study educational operating systems such as Minix (<http://www.minix3.org/>), Weenix (<http://weenix.cs.brown.edu/mediawiki/index.php/Weenix>) and develop reasonably sized interesting modules for them
10. CASE STUDY: Study the Android open source operating system for mobile devices (<http://source.android.com/>) and develop / modify some modules.
- 11 . CASE STUDY: Study any embedded and real-time operating system such as eCos (<http://ecos.sourceware.org/>) and develop / modify some modules.

REFERENCES:

1. M Singhal and NG Sivaratri, Advanced Concepts in Operating Systems, Tata McGraw Hill Inc.,2001
2. A.S. Tanenbaum, Distributed Operating system, Pearson Education Asia, 2001.
3. A.S. Tanenbaum, Modern Operating system, Prentice Hall, 3rd edition.
4. Real Time Operating System, Barr M.
5. Real-Time Systems, Jane Liu, Pearson Ed. Asia
6. Real -Time Systems, Krishna and Shin, McGraw Hill International.
7. Smart phone operating system concepts with Symbian O.S. A tutorial guide by Michael J. Jipping. Symbian Press, Wiley.
8. Application development using Android, Hello, Android, mobile development platform 3rd Edition by Ed Burnette
9. Silberschatz and P. Galvin, Operating System Concepts, VI edition, Addison Wesley 2004.

**ADVANCED DATABASES
(Only for CSE)**

L	T	P	C
3	2	1	5

Course objectives:

- To learn the modeling and design of databases.
- To acquire knowledge on parallel and distributed databases and its applications.
- To study the usage and applications of Object Oriented database
- To understand the principles of intelligent databases.
- To understand the usage of advanced data models.
- To learn emerging databases such as XML, Cloud and Big Data.
- To acquire inquisitive attitude towards research topics in databases.

PARALLEL AND DISTRIBUTED DATABASES: Database System Architectures: Centralized and Client-Server Architectures – Server System Architectures – Parallel Systems- Distributed Systems – Parallel Databases: I/O Parallelism – Inter and Intra Query Parallelism – Inter and Intra operation Parallelism – Design of Parallel Systems- Distributed Database Concepts - Distributed Data Storage – Distributed Transactions – Commit Protocols – Concurrency Control – Distributed Query Processing – Case Studies

OBJECT AND OBJECT RELATIONAL DATABASES: Concepts for Object Databases: Object Identity – Object structure – Type Constructors – Encapsulation of Operations – Methods – Persistence – Type and Class Hierarchies – Inheritance – Complex Objects – Object Database Standards, Languages and Design: ODMG Model – ODL – OQL – Object Relational and Extended – Relational Systems: Object Relational features in SQL/Oracle – Case Studies.

INTELLIGENT DATABASES: Active Databases: Syntax and Semantics (Starburst, Oracle, DB2)- Taxonomy- Applications- Design Principles for Active Rules- Temporal Databases: Overview of Temporal Databases- TSQL2- Deductive Databases: Logic of Query Languages – Datalog- Recursive Rules- Syntax and Semantics of Datalog Languages- Implementation of Rules and Recursion- Recursive Queries in SQL- Spatial Databases- Spatial Data Types- Spatial Relationships- Spatial Data Structures-Spatial Access Methods- Spatial DB Implementation.

ADVANCED DATA MODELS: Mobile Databases: Location and Handoff Management - Effect of Mobility on Data Management - Location Dependent Data Distribution - Mobile Transaction Models - Concurrency Control - Transaction Commit Protocols- Multimedia Databases- Information Retrieval- Data Warehousing- Data Mining- Text Mining.

EMERGING TECHNOLOGIES: DynamoDB vs MongoDB NoSQL - XML Databases: XML-Related Technologies-XML Schema- XML Query Languages- Storing XML in Databases-XML and SQL- Native XML Databases- Web Databases- Geographic Information Systems- Biological Data Management- Cloud Based Databases: Data Storage Systems on the Cloud- Cloud Storage Architectures-Cloud Data Models- Query Languages- Introduction to Big Data-Storage-Analysis.

LAB COMPONENTS (Only for CSE),

The students are expected to implement assigned lab exercises.

- Relational Databases: DDL, DML, Joins, Queries, Procedures and Functions, PL/SQL, Packages, Triggers
- ODBC connectivity
- Select the appropriate high performance database like parallel and distributed database
- Model and represent the real world data using object oriented database
- Design a semantic based database to meaningful data access
- Embed the rule set in the database to implement intelligent databases
- Represent the data using XML database for better interoperability
- Handle Big data and store in a transparent manner in the cloud
- To solve the issues related to the data storage and retrieval
- Other advanced database concepts

REFERENCES:

1. R. Elmasri, S.B. Navathe, –Fundamentals of Database Systems||, Fifth Edition, Pearson Education/Addison Wesley, 2007.
2. Thomas Cannolly and Carolyn Begg, –Database Systems, A Practical Approach to Design, Implementation and Management||, Third Edition, Pearson Education, 2007.
3. Henry F Korth, Abraham Silberschatz, S. Sudharshan, –Database System Concepts||, Fifth Edition, McGraw Hill, 2006.
4. C.J.Date, A.Kannan and S.Swamynathan, ||An Introduction to Database Systems||, Eighth Edition, Pearson Education, 2006.
5. Raghu Ramakrishnan, Johannes Gehrke, –Database Management Systems||, McGraw Hill, Third Edition 2004.

DATABASE ADMINISTRATION (Only for DBS)

L	T	P	C
3	0	0	3

Course Objectives: This subject will review the purpose and responsibilities of the Database Administrator. The student will learn how the DBMS manages the data and controls such as recovery, locking and transaction logging and performance tuning. Through hands on Database Administration using Oracle, the student will practice much of the theory presented.

DATABASE ADMINISTRATION: Creating database environment, Database design, application design, Design reviews.

DATABASE CHANGE MANAGEMENT: Data availability, performance management, system performance, database performance.

APPLICATION PERFORMANCE: Data integrity, database security, database backup and recovery, disaster planning

DATA AND STORAGE MANAGEMENT: data movement and distribution, database connectivity, Meta data management“

DBA TOOLS: DBA rules of thumb, DBA tool vendors, DBA web resources.

Case studies: Usage of DBA tools

REFERENCES

1. Database Administration The Complete Guide to Practices and Procedures, Craig S.Mullins, Addison Wisely
2. Understanding Database Administration, Pablo Berzukov.
3. Database Administration: The complete Guide to DBA practices and Procedures (2nd Edition), Craig S.Mullins, Addison Wisely Publications.

INFORMATION RETRIEVAL (Only for DBS)

L	T	P	C
3	0	0	3

Course Objectives: Information retrieval covers the tasks of indexing, searching, and recalling data, particularly text or other unstructured forms. It has an important role to play in a large number of applications viz., digital libraries, office automation, internet and e-commerce. The aim of the course is to study theoretical aspects as well as implementation issues of classical and modern retrieval problems.

EVALUATION OF INFORMATION REPRESENTATION AND RETRIEVAL - Information representation and retrieval overview, information representation basic approaches, information retrieval II (other topics). User dimension in information representation and retrieval.

LANGUAGE IN INFORMATION AND RETRIEVAL - Natural languages, controlled vocabulary, natural languages Vs controlled vocabulary, Language for IRR in the digital age.

RETRIEVAL TECHNIQUES AND QUERY REPRESENTATION – Retrieval representation, selection of retrieval techniques, query representation, retrieval of information unique in content or format.

RETRIEVAL APPROACHES- Retrieval by searching, retrieval by browsing, searching and browsing integrated in retrieval, Information retrieval models- Foundation of all IR Models, the Boolean logic model, vector space model, probability model, extensions of major IR models,.

INFORMATION RETRIEVAL SYSTEMS – online systems, CD-ROM Systems, OPACS, internet retrieval systems.

REFERENC ES

1. Information Representation and retrieval in the digital age, Heting Chu, Thomas H.Hogan, Sr, ASIST Monograph series, 2007.
2. Information Retrieval : Algorithms and Heuristic, Springer 1998.
3. Search Engines: Information Retrieval in Practice, W.Bruce Croft, Donald Metzler, Trevor Strohman, Addison Wesley Company, 2010.
4. Introduction to Modern information Retrieval, Gobinda G.Chowdry, Library Association Publishing, 1999.

SOFTWARE TESTING AND QUALITY ASSURANCE (Only for SE)

Course Objectives:

L	T	P	C
3	0	2	5

- To provide students with the knowledge, skill and in-depth technical understanding of the key concepts required to design and build dependable software systems.
- To achieve the gap between the theory and practice of software development and, in particular, through the transfer of knowledge from the classroom via research driven and commercially relevant project.

Introduction to Software Testing - Purpose of Testing - A model for testing - A taxonomy of Bugs - Software Safety - Software Testing Realities - Testing Software Specifications – The Need for Levels of Testing-Levels of Testing and Software Development Paradigms.

Black-box Testing – Test Case Design Strategies-Using the Black box Approach to Test Case Design- random Testing-Equivalence Class Partitioning-Boundary Value Analysis-An Example of the Application of Equivalence Class Partitioning and Boundary Value Analysis- Other Black Box Test Design Approaches-Cause and Effect Graphing-State Transition Testing- Error Guessing.

White-box Testing : Test Adequacy Criteria-Coverage and Control Flow Graphs-Covering Code Logic- Paths: Their Role in White Box-Based Test design-Additional White Box Test Design Approaches-Data Flow and White Box Test Design-Loop Testing-Mutation Testing and Program Perturbation.

Other testing techniques: Website Testing - Usability Testing - Code Inspections - Testing the Documentation -Testing for Security - Program Slicing - Syntax Testing - Object-Oriented Protocol Testing

Software Metrics: Linguistic Metrics - Structural Metrics - Hybrid Metrics - Metrics Implementation

Reliability: reliability metrics, fault avoidance, Cleanroom software development, fault tolerance, exception handling, N-version programming, recovery blocks, formal methods, functional specifications, and Z notation.

Software quality assurance: SQA basics-Software quality in business context – Planning for software quality assurance – Product quality and process quality – Software process models - Total Quality Management- 7 QC Tools and Modern Tools.

Quality assurance models : Models for Quality Assurance-ISO-9000 – Series- CMM- CMMI- Test Maturity Models, SPICE, Malcolm Baldrige Model- P-CMM.

Software quality assurance trends: Software Process- PSP and TSP - OO Methodology, Clean-room software engineering, Defect injection and prevention -Internal Auditing and Assessments-Inspections & Walkthroughs.

LAB COMPONENTS: (Only for SE)

The students are expected to carry out one or two case studies.

CASE STUDY 1: Cause Effect Graph Testing for a Triangle Program

Perform cause effect graph testing to find a set of test cases for the following program specification: Write a program that takes three positive integers as input and determine if they represent three sides of a triangle, and if they do, indicate what type of triangle it is. To be more specific, it should read three integers and set a flag as follows:

- If they represent a scalene triangle, set it to 1.
- If they represent an isosceles triangle, set it to 2.
- If they represent an equilateral triangle, set it to 3.
- If they do not represent a triangle, set it to 4.

CASE STUDY 2: Boundary Value Analysis for a Software Unit

The following is a specification for a software unit. The unit computes the average of 25 floating point numbers that lie on or between bounding values which are positive values from 1.0 (lowest allowed boundary value) to 5000.0 (highest allowed boundary value). The bounding values and the numbers to average are inputs to the unit. The upper bound must be greater than the lower bound. If an invalid set of values is input for the boundaries an error message appears and the user is reported. If the boundary values are valid the unit computes the sum and the average of the numbers on and within the bounds. The average and sum are output by the unit, as well as the total number of inputs that lie within the boundaries. Derive a set of equivalence classes for the averaging unit using the specification, and complement the classes using boundary value analysis. Be sure to identify valid and invalid classes.

Design a set of test cases for the unit using your equivalence classes and boundary values. For each test case, specify the equivalence classes covered, input values, expected outputs, and test case identifier. Show in tabular form that you have covered all the classes and boundaries. Implement this module in the programming language of your choice. Run the module with your test cases and record the actual outputs. Save an uncorrected version of the program for future use.

CASE STUDY 3: Cyclomatic Complexity for Binary Search

Draw a control flow graph for the given binary search code and clearly label each node so that it is linked to its corresponding statement. Calculate its cyclomatic complexity.

```
int binsearch (int x, int v[], int n)
{
    int low, high, mid;
    low = 0;
    high = n-1;
    while (low <= high) {
        mid = (low+high)/2
        if (x < v[mid]
            high = mid-1;
        else if (x > v[mid])
            low = mid+1;
        else /* found match */
```

```

return mid;
}
return -1; /* no match*/
}

```

CASE STUDY 4: Data Flow Testing for Gregorian Calendar

A program was written to determine if a given year in the Gregorian calendar is a leap year. The well-known part of the rule, stipulating that it is a leap year if it is divisible by 4, is implemented correctly in the program. The programmer, however, is unaware of the exceptions: A centenary year, although divisible by 4, is not a leap year unless it is also divisible by 400. Thus, while year 2000 was a leap year, the years 1800 and 1900 were not. Determine if the following test-case selection criteria are reliable or valid.

- (a) $C1(T) \equiv (T = \{1, 101, 1001, 10001\})$
- (b) $C2(T) \equiv (T = \{t \mid 1995 \leq t \leq 2005\})$
- (c) $C3(T) \equiv (T = \{t \mid 1895 \leq t \leq 1905\})$
- (d) $C4(T) \equiv (T = \{t\} \wedge t \in \{400, 800, 1200, 1600, 2000, 2400\})$
- (e) $C5(T) \equiv (T = \{t, t+1, t+2, t+3, t+4\} \wedge t \in \{100, 200, 300, 400, 500\})$
- (f) $C6(T) \equiv (T = \{t, t+1, t+2, \dots, t+399\} \wedge t \in D)$
- (g) $C7(T) \equiv (T = \{t1, t2, t3\} \wedge t1, t2, t3 \in D)$

CASE STUDY 5: State based Testing for an Assembler

Suppose you were developing a simple assembler whose syntax can be described as follows :

`<statement_> :: = <label field><op code><address>`

`<label field> :: = none | <identifier> :`

`<op code> :: = MOVE | JUMP`

`<address> :: = <identifier> | <unsigned integer>`

A stream of tokens is input to the assembler. The possible states for such an assembler are:

S1, prelabel; S2, label; S3, valid op code; S4, valid address; S5, valid numeric address. Start, Error, and Done. A table that describes the inputs and actions for the assembler is as follows:

Inputs

no more tokens

Identifier

MOVE, JUMP

colon

Integer

Actions

A1: Put the label in the symbol table.

A2: Look up the op code and store its binary value in op code field.

A3: Look up symbol in symbol table and store its value in address field.

A4: Convert number to binary, and store that value in address field.

A5: Place instruction in the object module, and print a line in the listing.

A6: Print error message and put all zeroes in the instruction.

Using this information and any assumptions you need to make, develop a state transition diagram for the assembler. From the state transition diagram develop a set of test cases that will cover all of the state transitions. Be sure to describe the exact sequence of inputs as well as the expected Sequence of state changes and actions.

CASE STUDY 6: Stress Testing of a Map-Aided Vehicle Tracking and Scheduling System

The American package courier and freight business faced the double pressures of consolidation and unstoppable increases in fuel costs. In mid-2008, pump prices were already double those

prevailing in early 2007. As well, the recent decision of long-time price leader DHL to –co-locate dozens of routes with erstwhile competitor UPS revealed just how fragile are market positions built through decades of promotions.

In Omaha, regional freight leader Red Ball Trucking¹ was keener than most to maximize operating efficiencies out of its substantial fleet of trucks and vans and thereby maintain margins in the face of low-cost rivals. In March 2008, a brand-new map-based adjunct to the company’s proprietary logistics and routing system neared rollout. Extensive –white box², line-by-line testing had eliminated most of the gross errors but the Red Ball CEO was concerned about the scalability of the program test bed.

Find out whether the map-enhanced vehicle tracking and scheduling system would remain stable at benchmarks of 50, 100 and 1000 concurrent users. Clean up any remaining bugs not caught by in-house.

CASE STUDY 7: Model Based Testing

Design and develop a scientific calculator program using various GUI components and events. Build the test model for the same. Determine the inputs that can be given to the model. Calculate expected output for the model. Run the test cases. Compare the actual output with the expected output. Any model based technique can be used for building the test model.

CASE STUDY 8: Web Application Testing for Student Grade System

With educational organizations under increasing pressure to improve their performance to secure funding for future provision of programmes, it is vital that they have accurate, up-to-date information. For this reason, they have MIS systems to record and track student enrolment and results on completion of a learning programme. In this way they can monitor achievement statistics. All student assignment work is marked and recorded by individual module tutors using a spreadsheet, or similar, of their own design. In the computing department these results are input into a master spreadsheet to track a student’s overall progress throughout their programme of study. This is then made available to students through the web portal used in college. Perform web application testing for this scenario.

CASE STUDY 9: Consider an Web based Loan application system <http://www.mortgagecalculator.org/> , which calculates the total interest, Monthly payment etc., to be paid based on the Loan term, loan interest, loan amount etc.,

- a) Write down the **requirement specification** for the above application.
- b) Design the **test cases** and perform **Validation testing** for the above application using any automated tool.

CASE STUDY 10: Consider an Web based Tax application <http://finotax.com/itax/itax1011.htm>, which calculates the Total tax to be paid for the academic year based on Total income, total deductions etc.,

- a) Write down the **requirement specification** for the above application.
- b) Design **test cases** and perform **Functional testing**.

CASE STUDY 11: Develop a Java program for student marks analyzing system.

- a) The System obtains following information’s from the faculty generates report Roll No, Name, Department, Semester, Marks obtained in each subject.
- b) The total for each student should be calculated and ranked based on total and pass in all the subject appeared.
- c) The Final report should display rank, percentage, Class, Pass/Fail Status for each

student.

Perform Unit testing for the above program using **Junit** feature.

CASE STUDY 12: Consider an Web based Loan application system <http://www.mortgagecalculator.org/>, which calculates the total interest, Monthly payment etc., to be paid based on the Loan term, loan interest, loan amount etc.,

- a) Write down the **requirement specification** for the above application.
- b) Design the **test cases** and perform **Function testing** for the above application using any automated tool.

CASE STUDY 13: Consider an Web based Tax application <http://finotax.com/itax/itax1011.htm>, which calculates the Total tax to be paid for the academic year based on Total income, total deductions etc.,

- a) Write down the **requirement specification** for the above application.
- b) Design **test cases** and perform **performance testing**, using any automated tool.

CASE STUDY 14: Consider the ticket reservation scenario of online Air ticket reservation system and perform Load testing and analyze its metrics like response time, through put, Maximum user load, resource utilization.

CASE STUDY 15: Consider the browse catalog scenario of Library management system and perform stress testing and verify whether the system is able to recover itself after stress testing.

CASE STUDY 16: Develop a java GUI for Login scenario of course registration system. Implement encryption technique in the above scenario.

- a) Perform manual security testing to crack the above program.
- b) Calculate the response time in the above scenario.

CASE STUDY 17: Write a Java program for sorting the input in ascending order.

- a) Calculate cyclomatic complexity
- b) Perform Manual testing to see whether all the paths are covered or not.
- c) Perform Manual testing to see whether all the decision braches are covered or not.

CASE STUDY 18: Consider an Web based Tax application <http://finotax.com/itax/itax1011.htm>, which calculates the Total tax to be paid for the academic year based on Total income, total deductions etc.,

- a) Write down the **requirement specification** for the above application.
- b) perform **Performance testing and measure its metrics**.

CASE STUDY 19: Develop a Java program for student marks analyzing system.

- a) The System obtains following information's from the faculty generates report Roll No, Name, Department, Semester, Marks obtained in each subject.
- b) The total for each student should be calculated and ranked based on total and pass in all the subject appeared.
- c) The Final report should display rank, percentage, Class, Pass/Fail Status for each student.

Perform Functionality testing using manual techniques.

CASE STUDY 20: Develop a Java program for student marks analyzing system.

- a) The System obtains following information's from the faculty generates report Roll No, Name, Department, Semester, Marks obtained in each subject.
- b) The total for each student should be calculated and ranked based on total and pass in all the subject appeared.
- c) The Final report should display rank, percentage, Class, Pass/Fail Status for each student.

Calculate response time and the memory used by the program.

CASE STUDY 21: Consider an purchasing of product scenario of online shopping system,

- a) Perform Validation testing of the GUI
- b) Perform Security testing using any automated tool.

CASE STUDY 22: Develop a Java program for student marks analyzing system.

- a) The System obtains following information's from the faculty generates report Roll No, Name, Department, Semester, Marks obtained in each subject.
- b) The total for each student should be calculated and ranked based on total and pass in all the subject appeared.
- c) The Final report should display rank, percentage, Class, Pass/Fail Status for each student.

Perform Integration testing for the above program.

CASE STUDY 23: Consider any Online Air reservation system website

Perform HTML code validation and CSS validation Testing using any automated tool.

CASE STUDY 24: Consider an online course registration system website. Perform Link checking and HTML validation using any automated tool.

CASE STUDY 25: Consider an online movie booking system. A customer can book any number of tickets either for the same day or different days. Customers can also modify or cancel the booked tickets. Payments are made through a secured gateway system. Design a webpage for booking the ticket scenario using HTML language and perform HTML validation.

CASE STUDY 26: Consider an online movie booking system. A customer can book any number of tickets either for the same day or different days. Customers can also modify or cancel the booked tickets. Payments are made through a secured gateway system. Design a webpage for booking the ticket scenario using HTML language and perform validation testing.

CASE STUDY 27: Design a simple online chat application which can transfer text messages using java. Calculate its response time.

CASE STUDY 28: Consider any online shopping system website. Perform Validation testing and functionality testing.

CASE STUDY 29: Prepare test plan and develop test case hierarchy

CASE STUDY 30: Generate Test cases and Test Documentation in the following applications Library System, Course Registration System, Implement a Quiz System, Student Marks Analyzing System, Online Ticket Reservation System, Stock Management System, Real- Time Scheduler

REFERENCE

S:

6. Boris Beizer, -Software Testing Techniques, 2nd Edition, Dream tech press, New Delhi, 2003.
7. Edward Kit, -Software Testing in the Real World - Improving the Process, Pearson Education, 2004.

8. William E. Perry, -Effective methods for software testing, 2nd Edition, John Wiley, 2000.
9. Stephen H.Kan, Metric and Models in software Quality Engineering, Addison QWesley 1995.
10. S.A.Kelkar, Software quality and Testing, PHI Learning, Pvt, Ltd., New Delhi 2012

OPEN SOURCE SOFTWARE (Only for SE)

L	T	P	C
3	0	0	3

Course Objectives:

1. To analyze and understand the differences between the open source model and the commercial proprietary model
2. To gain hands-on skills and experience in using a variety of open source software and to develop and use evaluative methods for assessing OSS products

LINUX BASICS: Overview of free/open source software – Definition of FOSS and GNU – History of GNU/Linux and the free software movement – Advantages of free software and GNU/Linux – FOSS usage – Trends and potential – Global and Indian – GNU/Linux OS installation – Detect hardware – Configure disk partitions and file systems – Install a GNU/Linux distribution – Basic shell commands – Logging in – Listing files – Editing files – Copying/moving files – Viewing file contents – Changing file modes and permissions – Process management – User and group management – File ownerships and permissions – PAM authentication – Introduction to common system configuration files and log files – Configuring networking – Basics of TCP/IP networking and routing – Connecting to the internet. (through dialup –DSL- Ethernet –Leased line)

LINUX FUNDAMENTALS: Configuring additional hardware – Sound cards – Displays and Display cards – Network cards – Modems – USB drives – CD writers – Understanding the OS bootup process – Performing everyday tasks using gnu/linux – Accessing the internet playing music – Editing documents and spreadsheets – Sending and receiving email – Copy files from disks and over the network – Playing games – Writing CDS – X window system configuration and utilities – Configure X windows – Detect display devices – Installing software – From source code as well as using binary packages – Setting up email servers – Using postfix (SMTP Services) – Courier (MAP & POP3 Services) – Squirrel mail (Web Mail Services) – Setting up web servers – Using apache (HTTP Services) – PHP (Server-Side Scripting) – Perl (CGI Support) – Setting up file services – Using samba (File and Authentication Services for Windows Networks) – Using NFS (File Services for Gnu/Linux / Unix Networks) – Setting up proxy services – Using squid (Http / Ftp / Https Proxy Services) – Setting up printer services – Using CUPS (Print Spooler) – Foomatic.(Printer Database)

DEVELOPMENT ENVIRONMENT: Setting up a firewall – Using netfilter and IP tables – Using the GNU compiler collection – GNU compiler tools – C preprocessor (CPP) – C compiler (GCC) and the C++ compiler (G++) – Assembler (GAS) – Understanding build systems – Constructing make files and using make – Using autoconf and autogen to automatically generate make files tailored for different development environments – Using source code versioning and management tools – Using CVS to manage source code revisions – Patch and diff.

LINUX INTERNALS: Understanding the GNU LIBC libraries and linker – Linking against object archives (.a libraries) and dynamic shared object libraries (.so libraries) – Generating statically linked binaries and libraries – Generating dynamically linked libraries – Using the GNU debugging tools – GDB to debug programs – Graphical debuggers like DDD – Memory debugging / profiling libraries MPATROL and VALGRIND – Review of common programming practices and guidelines for GNU/Linux and FOSS – Basics of bash – SED and Awk scripting- Basics of the X windows server architecture.

DESKTOP PROGRAMMING: QT programming – GTK+ programming – Python programming – Programming GUI applications with localization support.

OPEN STANDARDS: National Information Standards Organization (NISO), The Digital Library Federation (DLF). The Dublin Core Metadata Initiative. MARC standards, Resource Description and Access (RDA). Open Archives Initiative. OAI-PMH. Search / Retrieval via URL (SRU), SRW/CQL. Java Platform, Enterprise Edition (Java EE).

OPEN SOURCE LICENSES: GNU General Public License (GPL) version 2,3, GNU Lesser General Public License (LGPL) version 2.1,3, GNU Affero General Public License (AGPL) version 3, Apache License, Version 2.0, Artistic License 2.0, etc.

REFERENCES

ES

1. N. B. Venkateshwarlu, -Introduction to Linux: Installation and Programming, B S Publishers, 2005.
2. Matt Welsh, Matthias Kalle Dalheimer, Terry Dawson and Lar Kaufman, -Running Linux, 4th Edition, O'Reilly Publishers, 2002.
3. Carla Schroder, -Linux Cookbook, 1st Edition, O'Reilly Cookbooks Series, November 2004.
4. -Open Sources: Voices from the Open Source Revolution, 1st Edition, January 1999.
URL: <http://www.oreilly.com/catalog/opensources/book/toc.html>
5. The Linux Cookbook: Tips and Techniques for Everyday Use, 1st Edition, Michael Stutz, 2001. URL: http://dsl.org/cookbook/cookbook_toc.html
6. -The Linux System Administrators' Guide, Lars Wirzenius, Joanna Oja, Stephen Stafford, and Alex Weeks, December 2003. URL: <http://www.tldp.org/guides.html>
7. Using GCC, Richard Stallman et al. URL: <http://www.gnu.org/doc/using.html>
8. An Introduction to GCC, Brian Gough.
URL: <http://www.networktheory.co.uk/docs/gccintro/>
9. GNU Autoconf, Automake and Libtool, Gary V. Vaughan, Ben Elliston, Tom Tromey and Ian Lance Taylor. URL: <http://sources.redhat.com/autobook/>
10. Open Source Development with CVS, 3rd Edition, Karl Fogel and Moshe Bar.
URL: <http://cvsbook.red-bean.com/>
11. Advanced Bash Scripting Guide, Mendel Cooper, June 2005.
URL: <http://www.tldp.org/guides.html>
12. GTK+/GNOME Application Development, Havoc Pennington.
URL: <http://developer.gnome.org/doc/GGAD/>

13. Python Tutorial, Guido van Rossum, Fred L. Drake, Jr., Editor.
URL: <http://www.python.org/doc/current/tut/tut.html>
14. <http://directory.fsf.org/GNU/>
15. <http://www.diglib.org>
16. <http://www.niso.org/>
17. The Dublin Core Metadata Initiative <<http://dublincore.org/>>
18. MARC standards <<http://www.loc.gov/marc/>>
19. Resource Description and Access (RDA) <<http://www.rdaonline.org/>>
20. WAMP server (Windows, Apache, MySQL, PHP) <<http://www.wampserver.com/en/>>
21. Open Source MySQL <<http://www.mysql.com/>>
22. Search / Retrieval via URL (SRU) <<http://www.loc.gov/standards/sru/>>
23. <<http://www.loc.gov/standards/>>
24. <<http://iesr.ac.uk/use/sru/>>
25. <<http://java.sun.com/javaee/>>
26. <<http://www.jboss.org/>>

SOFTWARE MAINTENANCE AND EVOLUTION (Only for SE)

- To acquire an understanding and appreciation of the challenges posed by software maintenance and software evolution.
- To learn about quality of architecture and source code and how it affects software maintenance and evolution.

INTRODUCTION: Basic Concepts- The Maintenance Framework - Fundamentals of Software Change - Limitations and Economic Implications to Software Change - The Maintenance Process.

MAINTENANCE ACTIVITIES: Program Understanding - Reverse Engineering - Reuse and Reusability – Testing - Management and Organisational Issues.

MANAGEMENT: Configuration Management - Maintenance Measures.

BUILDING BETTER SYSTEMS: Building and Sustaining Maintainability – Maintenance Tools

STANDARDS: IEEE Standard for Software Maintenance

REFERENCeS:

1. Grubb and Takang, Software Maintenance: Concepts and Practices, 2nd edition, World Scientific, 2003.
2. Stanislaw Jarzabek, Effective Software Maintenance and Evolution: A Reuse-Based Approach, Auerbach publications, 2007.
3. April A. and Abran A., Software Maintenance Management, Wiley, 2008.- Guide to the Software Engineering Body of Knowledge: 2004 ed., Abran A., Moore J.W.(Executive Ed.), Bourque P., Dupuis R. (Ed.). IEEE Computer Society Press, April 2005.
4. IEEE Std 1219-1998, IEEE Standard for Software Maintenance.

ENTERPRISE SOFTWARE ARCHITECTURE AND DESIGN (Only for SE)

L	T	P	C
3	0	2	4

Course Objectives:

- To address the important engineering issues in building large-scale enterprise software systems.
- To emphasize service-oriented architectures (SOA) and best practices for building service-oriented enterprises in a vendor-neutral fashion.
- To understand the domain-driven architecture, service-oriented architecture and resource-oriented architecture.

Middleware: Enterprise Information Systems - Communication - System and Failure Models - Remote Procedure Call - Message-Oriented Middleware - Web Services and Service-Oriented Architecture (SOA) - Cloud Computing - Naming and Discovery.

Data Modeling: Entities and Relationships - Concepts and Entities - Attributes and Relationships - Properties of Relationship Types - Special Relationship Types - XML Schemas - Defining New Types - Defining Simple Types - Defining Complex Types - Derived Types - Derived Simple Types - Derived Complex Types - Document Hierarchies - Relationship Types in XML Schemas - Meta schemas and Meta models.

Data Processing: Processing XML Data - Tree Processing - Schema Binding - Stream Processing - External Processing - Query Languages and XQuery - XML Databases - Storage as Relational Tables - Storage as Large Strings - Native XML Storage - Web Services - Simple Object Access Protocol - WSDL: Web Services Description Language - Web Service Policy - Presentation Layer: JSON and JQUERY.

Domain-Driven Architecture: Software Architecture - Domain-Driven Design - Application Frameworks - Domain-Specific Languages (DSLs) - An Example API for Persistent Domain Objects - Domain-Driven Architecture.

Service-Oriented Architecture: Services and Procedures - Service-Oriented Architecture (SOA) - Service Design Principles - Service-Oriented Architecture (SOA) Governance - Standardized Service Contract - Operations Contract - Data Contract - Policy Contract - Binding Contract - Contract Versioning - Service Loose Coupling - Motivation for Loose Coupling - Contract Development - Loose Coupling Patterns - Cost of Loose Coupling - Service Abstraction - Platform Abstraction - Protocol Abstraction - Procedural Abstraction - State Abstraction - Data Abstraction - Endpoint Abstraction - Service Reusability - Parameterization and Bounded Polymorphism - Subtyping, Inheritance, and Contracts - Does Service-Oriented Architecture Require Subtyping - Patterns for Service Reusability - Service Autonomy - Replicating Computation - Replicating State - Sources of Errors and Rejuvenation - Caching - Service Statelessness - Contexts and Dependency Injection - Service Discoverability - Global Discovery - Local Discovery - Layered Naming - Further Patterns.

Resource-Oriented Architecture: Representational State Transfer - RESTful Web Services - Resource-Oriented Architecture (ROA) - Interface Description Languages - Web Services Description Language (WSDL) - Web Application Description Language (WADL) - An Example Application Program Interface (API) for Resource-Oriented Web Services - Hypermedia Control and Contract Conformance.

LAB COMPONENTS: (Only for SE)

The students are expected to carry out one or two case studies.

CASE Study 1: Create a web page with the following using HTML5

- (i) To embed an image map in a web page
- (ii) To fix the hot spots
- (iii) Show all the related information when the hot spots are clicked.

CASE Study 2: Create a web page with all types of Cascading style sheets.

CASE Study 3: Implement Client Side Scripts for Validating Web Form Controls using JavaScript.

CASE Study 4: Designing Quiz Application Personal Information System Using JavaScript

CASE Study 5: Write a JavaScript for Loan Calculation.

CASE Study 6: Develop and demonstrate a HTML file that includes JavaScript that uses functions for the following problems:

- a) Parameter: A string
Output: The position in the string of the left-most vowel
- b) Parameter: A number
Output: The number with its digits in the reverse order

CASE Study 7: Develop PHP program using Arrays, control structures, looping structures and Form Handling

CASE Study 8: Using PHP and MySQL, develop a program to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

CASE Study 9: Write an AJAX program for parsing a JSON file and formatting the output.

CASE Study 10: Develop a web application for Airline Reservation System using PHP & AJAX.

CASE Study 11: Develop a web application for Airline Reservation System using EJB.

CASE Study 12: Develop a web application for Airline Reservation System using JSP.

CASE Study 13: Writing an RMI application for Airline Reservation System to access a remote method

CASE Study 14: Writing an application for Airline Reservation System using CORBA

CASE Study 15: Writing an application for Airline Reservation System using COM

CASE Study 16: Writing an application for Airline Reservation System using DCOM

CASE Study 17: Writing a Servlet program with database connectivity for a web based application such as students result status checking, PNR number enquiry etc

CASE Study 18: XML document creation.

CASE Study 19: Importing and Exporting XML document in database.

CASE Study 20: XSL Transformation

CASE Study 21: Internal and External DTD creation

CASE Study 22: XML Schema creation

CASE Study 23: Parsing XML document using DOM/SAX parser.

CASE Study 24: Web Service creation using JAX-WS
CASE Study 25: Web Service creation using JAX-RS
CASE Study 26: Web Service creation using .NET **CASE**
Study 27: JAXB Marshaling and Unmarshaling **CASE**
Study 28: Implementation of RESTful services **CASE**
Study 29: Design of simple cloud services

REFERENCES:

1. Dominic Duggan , Enterprise Software Architecture and Design , Wiley, 2012
2. Dominic Duggan , Enterprise and Cloud Computing , Wiley, 2013.