P14SE101 DISCRETE MATHEMATICS & OPTIMIZATION TECHNIQUES


Teaching Scheme:

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

<table>
<thead>
<tr>
<th></th>
<th>Continuous Internal Evaluation</th>
<th>40 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>End Semester Exam</td>
<td>60 marks</td>
</tr>
</tbody>
</table>

Course Learning Objectives:

- To introduce the methods of optimization of both linear and non-linear objectives under a set of constraints.
- To introduce the techniques of solving decision making problems and analyze them in a competitive situations to get optimal output.
- To introduce the concepts and determination of optimal flow in a transport network and analysis of network scheduling by CPM-PERT with their practical applications.
- To introduce the basic concepts of Fuzzy sets, Fuzzy operations, Fuzzy logic and their Engineering applications.

UNIT – I (9+3)


UNIT – II (9+3)

Decision Analysis and Game Theory: Introduction to decision making problems, Decision making under uncertainty, Laplace criterion, Max-min criterion, Savage criterion and hurwitz criterion, Introduction to game theory, Games with pure strategies. Max-min and min- max principle, Optimal solution of two person zero-sum game, Dominance property, Solutions of mixed strategy games using graphical and linear programming methods.

UNIT-III (9+3)

UNIT-IV (9+3)

Fuzzy Sets and Fuzzy Logic: Basic concepts of fuzzy set and examples, Operations on fuzzy sets, Fuzzy complements, Fuzzy intersections, Fuzzy union and their properties, \( \alpha \)-cuts and representation fuzzy sets, Generalized fuzzy operations, Complement, t-norms and T-Conorms, Simple theorems on fuzzy operations, Basic concepts of fuzzy logic, Fuzzy propositions and types of fuzzy propositions, Fuzzy quantifiers, Inferences from conditional fuzzy propositions, Qualified propositions and quantified propositions.

TEXT BOOKS:


REFERENCE BOOKS:


<table>
<thead>
<tr>
<th>Course Learning out Comes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>After attending the course the student will be able to</td>
</tr>
<tr>
<td>• Solve any type of LPP and discuss the nature of the solution.</td>
</tr>
<tr>
<td>• Solve a class of non-linear programming problems with different types of constraints.</td>
</tr>
<tr>
<td>• Identify the importance of decision making systems and find an optimal solution of the problem given different types of nature of states.</td>
</tr>
<tr>
<td>• Analyze different strategies of a Game between two objects under conflicting situations.</td>
</tr>
<tr>
<td>• Develop an algorithm for solving problems of Game theory.</td>
</tr>
<tr>
<td>• Find a maximal flow of commodities in a transport network using different methods.</td>
</tr>
<tr>
<td>• Discuss different network based methods designed to assist in the planning, scheduling and control of projects.</td>
</tr>
<tr>
<td>• Identify the differences between Crisp sets and Fuzzy sets and the related properties.</td>
</tr>
<tr>
<td>• Differentiate between Classical systems and Fuzzy systems in order to solve the problems based on Fuzzy logic.</td>
</tr>
</tbody>
</table>
P14SE102  OBJECT ORIENTED SOFTWARE ENGINEERING


Course learning objectives:
- To make perfect the students in modeling the systems.
- To understand project requirements elicitation.
- To develop and design a simplified system with reduced complexity.
- To understand the various testing methods.

UNIT - I (9+3)

UNIT - II (9+3)

UNIT - III (9+3)

UNIT - IV (9+3)
specification concepts, Interface specification activities, Managing object design, **Mapping Models to Code, Introduction:** A book example, An overview of mapping, Mapping concepts, Mapping activities and managing implementation, Testing, **Introduction:** Testing the space shuttle, An overview of testing, Testing concepts, Testing activities, Managing testing.

**TEXT BOOKS:**

**REFERENCE BOOKS:**

**Course learning outcomes:**
After completion of the course, the student will be able to

- model the systems effectively.
- elicit the project requirements.
- design the system in a simplified and understandable.
- test the systems effectively using appropriate testing methods.
M.Tech. Semester: I

Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

- Continuous Internal Evaluation: 40 marks
- End Semester Exam: 60 marks

Course Learning Objectives:
- To make students understand the knowledge of software requirements elicitation
- To improve students capability in defining project objectives
- To make the students develop appropriate design solutions to a given problem
- To improve students capability in developing quality software artifacts and that should be satisfied by the client

UNIT-I (9+3)

**Software Requirements:** Essential software requirement, Good practices for requirements engineering, Improving requirements processes, Software requirements and risk management, **Software Requirements Engineering:** Requirements elicitation, Requirements analysis documentation, Review, Elicitation techniques, Analysis models, Software quality attributes, Risk reduction through prototyping, Setting requirements priorities, Verifying requirements quality.

UNIT-II (9+3)

**Software Requirements Management:** Requirements management principles and practices, Requirements attributes, Change management process, Requirements traceability matrix, Links in requirements chain, **Software Requirements Modeling:** Use case modeling, Analysis models, Data flow diagram, State transition diagram, Class diagrams, Object analysis, Problem frames.

UNIT III (9+3)

**Software Estimation:** Components of software estimations, Estimation methods, Problems associated with estimation, Key project factors that influence estimation.

**Size Estimation:** Two views of sizing, Function point analysis, Mark II FPA, Full function points, LOC estimation, Conversion between size measures.

UNIT-IV (9+3)

**Effort, Schedule and Cost Estimation:** Productivity, Estimation factors, Approaches to effort and schedule estimation, COCOMO II, Putnam estimation model, Algorithmic models, Cost estimation, **Requirements and Estimation Management Tools:** Benefits of using a requirements management tool, commercial requirements management tool, Rational requisite pro, Caliber requirements management, Implementing requirements management automation, **Software Estimation Tools:** Desirable features in software estimation tools, International function point users group, USC’s COCOMO II, Software life cycle management tools.
Text Books

Reference Books

Course Learning Outcomes:
After completion of the course, the students will be able to

- model, analyze and measure the software artifacts
- analyze, specify and document software requirements for a software system
- verify, validate, assess and assure the quality of software artifacts
- understand the impact of computing solutions in a global and societal context
P14SE104  ADVANCED DATA STRUCTURES AND ALGORITHMS


Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

- Continuous Internal Evaluation: 40 marks
- End Semester Exam: 60 marks

Course Learning Objectives:

- To implement various operations on linear and non-linear data structures.
- To apply the suitable data structure to implement various sorting and searching techniques.
- To analyze the performance of different algorithms in terms of space and time.
- To understand various algorithm design methods and their usage in solving real world problems.

UNIT-I (9+3)

**Algorithms:** Definition, Properties, **Performance Analysis:** Time complexity and space complexity, Asymptotic notations, **Data Structures:** Definition, Linear and non linear data structures, Abstract data type concept, **Trees:** Basic terminology, Binary search trees, Traversal methods, AVL trees, Splay trees, Red-black trees, Skip lists, **Graphs:** Graphs terminology, Representations, Graph traversals methods –Depth first search and breadth first search.

UNIT-II (9+3)

- **Searching:** Linear and binary search methods, **Sorting:** Insertion sort, Heap sort and radix sort, **Internet Algorithms:** Strings and pattern matching algorithms, Cryptographic computations, Information security algorithms and Protocols, Network algorithms-Complexity measures and models, fundamental distributed algorithms, Broadcast and unicast routing, multi cast routing.

UNIT-III (9+3)

**Algorithm Design Methods:** Introduction, **Divide and Conquer:** General method, Merge sort, Quick sort, Sets and Disjoint sets, **Greedy method:** General method, Optimal storage on tapes, Knapsack problem, Minimum spanning trees, **Dynamic Programming:** Multistage graphs, Optimal binary search trees, Traveling sales person problem.

UNIT-IV (9+3)

**Back Tracking:** General method, 8- queens’s problem, Graph coloring problem, **Branch and Bound:** Introduction, 0/1 knapsack problem, Traveling sales person problem, Non-Polynomial-Hard and Non-Polynomial Complete Problems: Basic concepts, Nondeterministic algorithms, the classes NP-Hard and NP-Complete, Cook’s theorem.
Text Books:

Reference Books:

Course Learning Outcomes:
After completion of the course, the student will be able to

- know various linear and non-linear data structures, their operations and applications.
- analyze the performance of different algorithms in terms of space and time.
- implement various sorting and searching algorithms efficiently.
- select appropriate algorithm design method to solve a given real time problem.
P14SE105A  SECURE SOFTWARE ENGINEERING

M.Tech Semester: I

Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

- Continuous Internal Evaluation: 40 Marks
- End Semester Exam: 60 Marks

Course Learning Objectives:
- To make students capable of understanding the specification and design of secure software.
- To make students capable of developing secure software.
- To make students capable of testing security levels of an software.
- To make students capable of managing secure software's.

UNIT-I (9+3)

**Software Security Issues:** introduction, the problem, Software Assurance and Software Security, Threats to software security, Sources of software insecurity, Benefits of Detecting Software Security, **Secure Software Properties:** Properties of Secure Software, Influencing the security properties of Software, Asserting and specifying the desired security properties.

UNIT-II (9+3)

**Requirements engineering for secure software:** Introduction, the SQUARE process Model, Requirements elicitation and prioritization, **Secure Software Architecture and Design:** Introduction, software security practices for architecture and design, Architectural risk analysis.

UNIT-III (9+3)

**Knowledge for secure software design:** security principles, security guidelines and attack patterns. **Secure coding and Testing:** Code analysis, Software Security testing, Security testing, Considerations throughput the SDLC.

UNIT –IV (9+3)

**Secure Systems Assembling Challenges:** introduction, security failures, functional and attacker perspectives for security analysis, system complexity drivers and security, **Managing Secure Software’s:** Governance and security, Adopting an enterprise software security framework, Deciding how much security is enough, Security and project management, Maturity of Practices.
Text Books:


Reference books:


Course Learning Outcomes:

After the completion of course, the student will be able to:

- understand the specification and design of secure software.
- develop secure software
- test security levels of an software
- managing secure software’s
P14SE105B  COMPONENT BASED SOFTWARE ENGINEERING


Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:
- To expose the students to the concepts of component-based software engineering
- To make students capable of using software engineering practices for component building
- To make students capable of manage component based software systems.
- To make students capable of understanding real-time component technologies

UNIT I (9+3)
Component Introduction: Definition of a software component and its elements, The component industry metaphor, Component models and component services, An example specification for implementing a temperature regulator software component, The Case for Components: The business case for components, COT’S myths and other lessons learned in component-based software development.

UNIT II (9+3)

UNIT III (9+3)
The Management of Component-Based Software Systems: Measurement and metrics for software components, Implementing a practical reuse program for software components, Selecting the right COT’S software, Building instead of buying, Software component project management, The trouble with testing components, Configuration management and component libraries, The evolution, Maintenance and management of component based systems.

UNIT IV (9+3)
Component Technologies: Component Technologies, Overview of the CORBA component model, Overview of COM+ component model, Overview of the EJB component model, Bonobo and free software GNOME components, Choosing between COM+, EJB, and GNOME, Software Agents as Next Generation Software Components.
TEXT BOOKS:

REFERENCE BOOKS:

Course Learning Outcomes:

After completion of the course, the student will be able to

- know essentials concepts component-based software engineering
- apply software engineering practices for component-based systems
- manage projects of component based software systems.
- utilize the real-time component technologies in software building
P14SE105C SOFTWARE PROJECT MANAGEMENT


Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:

- To make students capable of understanding project management concepts and principles
- To make students capable of selecting the appropriate project development approach.
- To make students capable of performing costing and estimation of projects.
- To make students capable of performing risk assessment of projects.

UNIT – I (9+3)

Introduction to Software Project Management: Introduction to project, Project versus product, Product versus process, Software projects versus other types of project, Contract management and technical project management, Activities covered by software project management, Plans, methods and methodologies, Some ways of categorizing software projects, The Management spectrum, Problems with software projects, Setting objectives, Stakeholders, The business case, Requirement specification, Management control, Overview of Project Planning: Introduction to step wise project planning, Select project, Identify project scope and objectives, Identify project infrastructure, Analyze project characteristics, Identify project products and activities, Estimate effort for each activity, Identify activity risks, Allocate resources, Review/publicize plan, Execute plan and lower levels of planning, Project Evaluation: Strategic assessment, Technical assessment, Cost-benefit analysis, Cash flow forecasting, Cost-benefit evaluation techniques, Risk evaluation.

UNIT – II (9+3)

UNIT – III (9+3)

Risk Management: The nature of risk, Types of risk, Managing risk, Hazard identification, Hazard analysis, Risk planning and control, Evaluating risks to the schedule, Resource Allocation: The nature of resources, Identifying resource requirement, Scheduling resources, Creating critical paths, Counting the cost, Being specific, Publishing the resource schedule, Cost Schedules, The scheduling sequence, Monitoring and Control: Creating the framework, Collecting the data, Visualizing progress, Cost monitoring, Earned value, Prioritizing monitoring, Getting the project back to target, Charge control.

UNIT – IV (9+3)


TEXT BOOKS:

REFERENCE BOOKS:

Course Learning Outcomes:
After completion of the course, the student will be able to

- understand project management concepts and principles
- select the appropriate project development approach.
- perform costing and estimation of projects.
- perform risk assessment of projects.
P14SE105D  SERVICE ORIENTED ARCHITECTURE


Teaching Scheme :
<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:
- To expose the students to the basic principles of service oriented architecture
- To make students aware of Web service specifications and standards
- To make students capable of building service oriented web applications
- To make students capable of applying service layers in developing web services

Unit I (9+3)

Unit II (9+3)

Unit III (9+3)
Software Oriented Architecture and Services Orientation: Principles of service orientation, Service orientation and the enterprise, Anatomy of SOA, Common principles of service orientation, interrelation between principles of service, Orientation, Service orientation and object Orientation, Native web services support for principles of service orientation. Service layers, Service orientation and contemporary SOA, Service layer abstraction, Application service layer, Business service layer, Orchestration service layer, Agnostic services, Service layer configuration scenarios.
Unit IV (9+3)

Building Software Oriented Architecture: SOA delivery strategies, SOA delivery lifecycle phases, The top-down strategy, The bottom-up strategy, The agile strategy, Service oriented analysis, Introduction to service oriented analysis, Benefits of a business centric SOA, Deriving business services, Service oriented analysis, Service modeling, Service modeling guidelines, Classifying service model logic, Contrasting service modeling approaches.

TEXT BOOKS:

REFERENCE BOOKS:
2. N. M. Josuttis, SOA in Practice SPD.
3. M. Rosen and others, Applied SOA, Wiley India pvt. Ltd.

Course Learning Outcomes:
After completion of the course, the student will be able to

- understand basic principles of service oriented architecture
- gain knowledge on web service specifications and standards
- build service oriented web applications
- apply service layers in developing web services
P14SE106A  HUMAN COMPUTER INTERACTION


Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>40 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>60 marks</td>
</tr>
</tbody>
</table>

Course Learning Objectives:

- To expose students to the concepts, terminology, facts and principles in HCI.
- To know the relationships between specific instances and broader generalizations.
- To use concepts and principles to explain, analyze and solve specific situations.
- To of applying course content in coping with real life situations.

UNIT-I (9+3)

Introduction: Importance of user Interface Definition, Importance of good design, Benefits of good design, A brief history of screen design, The Graphical User Interface: Popularity of graphics, The concept of direct manipulation, Graphical system, Characteristics, Web user, Interface popularity, Characteristics, principles of user interface.

UNIT-II (9+3)

Design Process: Human interaction with computers, Importance of human characteristics human consideration, Human interaction speeds and understanding business junctions, Screen Designing: Design goals, Screen planning and purpose, Organizing screen elements, Ordering of screen data and content, Screen navigation and flow, Visually pleasing composition, Amount of information, Focus and emphasis, Presentation information simply and meaningfully, Information retrieval on web, Statistical graphics, Technological consideration in interface design.

UNIT-III (9+3)

Windows: New and navigation schemes selection of window, Selection of devices based and screen based controls, Components: Text and messages, Icons and increases, Multimedia, Colors, Uses problems, Choosing colors.

UNIT-IV (9+3)

Text Books:

Reference Books:

Course Learning Outcomes:
After completion of the course, the student will be able to

- know the basics of human and computational abilities and limitations.
- understand basic theories, tools and techniques in HCI.
- learn the fundamental aspects of designing and evaluating interfaces.
- practice a variety of simple methods for evaluating the quality of a user interface.
- apply appropriate HCI techniques to design systems

P14 SE106B  ADVANCED OPERATING SYSTEMS

M.Tech. Semester: I  
Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning objectives:

- To make students capable of understanding deadlocks and its recovery in distributed environment
- To make students known about load distribution requirements and algorithms
- To make students capable of system resource management and utilization
- To make students capable of understanding multiprocessor and data base operating systems

UNIT-I (9+3)

Process Synchronization: Introduction, Functions of an operating system, Design Approaches,  

UNIT – II (9+3)

Distributed Deadlock Detection: Preliminaries, Deadlock handling strategies in distributed systems, Issues in deadlock detection and resolution, Control organizations for distributed deadlock detection, Centralized deadlock detection algorithms, Distributed deadlock detection algorithms, Hierarchical deadlock detection algorithms,


UNIT-III (9+3)

Distributed Shared Memory: Architecture, Motivation, Algorithms for implementing DSM, Memory coherence, Coherence protocols, Distributed Scheduling: Issues in load distributing, Components of a load distributing algorithm, Load distributing algorithms, Task migration,  
UNIT - IV (9+3)

**Multiprocessor Operating Systems:** Motivations for multiprocessor systems, Basic multiprocessor System architectures: Tightly coupled versus loosely coupled, Uniform memory access Vs nonuniform memory Access Vs no remote memory access, **Interconnection Networks for Multiprocessor Systems:** Bus, Cross bar switch and multi stage inter connection network, Hyper cube architectures, **Case Studies:** The mach operating system, The sequoia system, **Database Operating Systems:** Introduction to database operating systems, Requirements of a database operating, **Concurrency Control:** The problem of concurrency control, Serializability Theory: Logs, Serial logs, Log equivalence, **Distributed Database Systems:** Data replication, Complications due to data replication, **Concurrency Control Algorithms:** Lock based algorithms, Timestamp based algorithms, Optimistic algorithms.

**TEXT BOOK:**


**REFERENCE BOOKS:**


**Course learning outcomes:**

- After the completion of course, the student will be able to
- understand deadlocks and its recovery in distributed environment
- known about load distribution requirements and algorithms
- perform system resource management and utilization
- understand multiprocessor and data base operating systems
P14SE106C REAL-TIME SYSTEMS


Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:

- To know the concept of a real-time systems
- To know the role and the design process of a real-time operating system
- To know the using generic process architectures for monitoring control and data acquisition systems

UNIT-I (9+3)


UNIT-II (9+3)

Commonly Used Approaches to Real-Time Scheduling: Clock driven approach, Weighted round-robin approach, Priority driven approach, Dynamic versus static systems, Effective release times and deadlines, Optimality of the EDF and LST algorithms, Non-optimality of the EDF and the LST algorithms, Challenges in validating timing constraints in priority-driven systems, Off-line Vs On-line scheduling.

Clock-Driven Scheduling: Notations and assumptions, Static timer-driven scheduler, General structure of cyclic schedules, Cyclic executives, Improving the average response time of periodic jobs, Scheduling sporadic jobs, Practical considerations and generalizations, Algorithms for constructing static schedules, Pros and cons of clock-driven scheduling, Priority driven scheduling of periodic task: Static assumption, Fixed-priority versus dynamic priority algorithms, Maximum schedulable utilization, Optimality of the RM and DM algorithms, A schedulability test for fixed-priority tasks with short response times, Schedulability test for priority tasks with arbitrary response times. Sufficient schedulability conditions for the RM and DM algorithms.
UNIT-III (9+3)

Scheduling Periodic and Sporadic Jobs In Priority-Driven System: Assumptions and approaches, Deferrable servers, Sporadic servers, Constant utilization, Total bandwidth and weighted fair -queuing server, Slack stealing in deadline-driven systems, Slack stealing in fixed priority system, Scheduling of sporadic jobs, Real-time performance for jobs with soft timing constraints, A two-level scheme for integrated scheduling.

Resources and Resource Access Control: Assumptions on resources and their usage, Effects of resource contention and resource access control, Non-preemptive critical sections, Basic priority-inheritance protocol, Basic priority-ceiling protocol, Stack-based, Priority-ceiling protocol, Use of priority-ceiling protocol in dynamic-priority systems, Preemptive-ceiling protocol, Controlling access to multiple-unit resources, Controlling concurrent access to data objects.

UNIT-IV(9+3)


TEXT BOOK:

REFERENCE BOOK:

Course Learning Outcomes:

After completion of the course, the student will be able to

- know the fundamental concepts in applications of computer science
- apply knowledge in advanced computer science to formulate the analyze problems in computing and solve them
- apply knowledge to the design and conduct experiments as well as to analyze and interpret data
- gain knowledge on emerging concepts in theory and applications of computer science
P14 SE106D       INFORMATION SYSTEMS AND AUDITING


Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:
- *This course helps to learn minute of information systems auditing organization and evaluating the different phases in systems development measures.*
- *It provides better understanding on security Mechanisms quality assures management and controls.*
- *It drives students to work on communication controls and processing controls.*
- *It gives better perception of concurrent auditing techniques and evaluating the system in an efficient way.*

UNIT-I (9+3)


UNIT-II (9+3)

UNIT-III (9+3)

**Input Controls:** Data input methods, Source document design, Data entry screen design, Data code controls, Check digits, Batch controls, Validation of data input, Instruction input, Validation of instruction input, Audit trail controls, Existence controls, **Communication Controls:** Communication subsystem exposures, Physical component controls, Line error controls, Flow controls, Link controls, Topological controls, Channel accesses controls, Controls over subversive threats, Internetworking controls, Communication architectures and controls, Audit trail controls, Existence controls, **Processing Controls:** Processor controls, Real memory controls, Virtual memory controls, Operating system integrity, Application software controls, Audit trail controls, Existence controls, **Database Controls:** Access controls, Integrity controls, Application software controls, Concurrency controls, Cryptographic controls, File handling controls, Audit trail controls, Existence controls, **Output Controls:** Inference controls, Batch output production and distribution controls, Batch report design controls, Online output production and distribution controls, Audit trail controls, Existence controls.

UNIT-IV (9+3)

**Audit Software:** Generalized audit software, Industry-specific audit software, High-level languages, Utility software, Expert systems, Neural network software, Specialized audit software control audit software, **Code Review, Test Data and Code Comparison:** Program source code review, Test data, Program code comparison, **Concurrent Auditing Techniques:** Basic nature of concurrent auditing techniques, Need for concurrent auditing techniques, Types of concurrent auditing techniques, Implementing concurrent auditing techniques, Strengths/limitations of concurrent auditing techniques, **Evaluating System Efficiency:** The evaluation process, Performance indices, workload models, System models, **Managing the Information System Audit Function:** Planning function, Organizing function, Staffing function, Leading function, Controlling function.

Text Book:

Reference Book:

**Course Learning Out Comes:**

*After completion of the course, the student will be able to*
- get knowledge on the information systems auditing and different audit procedures.
- understands the security management, operations management and quality assurance management controls for organizational issues.
- realize the process controls and database controls in information auditing.
- acquire insights on audit software and code review mechanism.
PI4SE107  OBJECT ORIENTED SOFTWARE ENGINEERING LABORATORY


Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam               | 60 marks |

Course Learning Objectives:

- To understand modeling the systems.
- To learn various UML diagrams.
- To expose effectively using UML diagrams in object oriented software design.
- To know the importance of case tools in software development and maintenance.

The following Experiments are suggested in this laboratory.

List of Experiments:

1. Developing Use-Case Analysis.
2. Developing Class & Object Diagrams.
5. Developing Component & Deployment Diagrams.
7. Case study on simple watch system & Library Information system.
8. Case study on Railway Reservation System & 2-floor elevator simulator system.
10. Case study on Hospital Management System & Online Shopping System.

Course Learning Outcomes:

After completion of the course, the student will be able to,

- model the system.
- design the object oriented software systems effectively using UML diagrams.
- use case tools effectively.
M.Tech. Semester: I
Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Examination Scheme:

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>40 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>60 marks</td>
</tr>
</tbody>
</table>

Course Learning Objectives:

- To learn windows applications in .net programming.
- To learn working with the data stored in databases in .net programming.
- To know the implementation of windows services and web services in .net programming.
- To create various web forms using Java scripts and Java Server Pages.
- To develop programs using java beans.

PART-A

List of experiments on .Net Programming:

1. Program to implementation of scientific calculator.
2. Program to implement the bouncing ball.
3. Program to draw Circle, Rectangle, Line, Ellipse and to fill them.
4. Program for creation of common dialog controls.(Open, Save, Font, Color).
5. Program to scroll the image using Scrollbars.
6. Program to handle the printer operations.
7. Program to Read/Write from/to Text files and Binary files.
8. Program to create a web form using the validation controls.
9. Program for create the Database table and perform the following operations.
   i) Insertion  ii) Deletion  iii) Updation  iv) Editing
10. Write the above program in Web Application.
11. Program to Synchronize the threads.
12. Program to create a Windows service and a Web service.
PART-B

List of experiments on Java 2 Enterprise Edition (J2EE):

1. Write a JSP program for displaying Employee details in a tabular format.
2. Write a JSP program to generate the following Employee form

![Employee Details Form]

Provide validations using Java Script for

1) All fields are mandatory
2) Empno, age, salary should be numeric
3) Age should be between 20 and 30
4) Employee name should be alphanumerical and should start with an upper case letter.

3. Design a form as follows

![Student Details Form]

i) Provide validations using Java Script
ii) Provide multiple buttons and call appropriate JSP files.
4. Design form as follows

![Login Form](image)

- Provide all validations
- Display appropriate messages like
- “welcome to user” if user id exists and password is correct
- “Welcome password” if user id exists and password is wrong
- “Invalid user” if user id does not exist
- Create a login table in oracle and connect to it.

5. Create a Bean for displaying welcome message, Invoke the Bean from JSP.

6. Create a Bean for implementing Account operations.

   Account Consists:
   - Datamembers: Accno, Balance, Account Type
   - Methods: Deposit, withdraw, getBalance

   - Write a JSP program which invokes the Account Bean and provide interface for invoking the methods of it.

7. Write a stateless Session Bean for accepting a string and returns “Welcome to ” followed by the accepted string.

8. Write a stateless Session Bean, which provides a remote interface consisting following interfaces.
   - void store(int a, int b)
   - int add();
   - int mul();

9. Write implementation file which implements the above methods.

10. Write a Client program for locating Session object and invokes the above methods.

11. Write a Stateless Session Bean for the above problem and observe the difference.

12. Create an Entity Bean which implements the Account Entity.
   - Data Members:
     - accountNo, balance, accountType.
   - Data Methods:
     1. void deposit(double amt);
     2. void withdraw(double amt);
     3. double getBalance();
13. Use Bean Managed Persistence as persistent-type.
14. Create an Entity Bean which Implements the Account Entity as above problem by using Container Managed Persistence as persistent-type

<table>
<thead>
<tr>
<th>Course Learning Outcomes:</th>
</tr>
</thead>
<tbody>
<tr>
<td>After completion of the course, the student will be able to</td>
</tr>
<tr>
<td>• write .Net programs to develop windows applications.</td>
</tr>
<tr>
<td>• establish the connection with the database in .Net programming.</td>
</tr>
<tr>
<td>• implement web services and windows services.</td>
</tr>
<tr>
<td>• to create web forms Java scripts and JSP.</td>
</tr>
<tr>
<td>• develop java beans for the user requirements.</td>
</tr>
</tbody>
</table>
P14SE109  SEMINAR

Class: M.Tech I Semester

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
</tbody>
</table>

Specialization: Software Engineering

Examination Scheme:

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>100 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>--</td>
</tr>
</tbody>
</table>

Guidelines:

The Department Post Graduate Review Committee (DPGRC) shall be constituted with HoD as a Chairman, M.Tech. Coordinator as a Convener and Three to five other faculty members representing various specializations in that particular programme as members. There shall be only Continuous Internal Evaluation (CIE) for Seminar, which includes Report Submission & Presentation.

A teacher will be allotted to a student for guiding in
(i) Selection of topic
(ii) Report writing
(iii) Presentation (PPT) before the DPGRC
P14SE201 SOFTWARE ARCHITECTURE AND DESIGN PATTERNS

M.Tech. Semester: II

Specialization: Software Engineering

Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation : | 40 marks |
| End Semester Exam : | 60 marks |

Course Learning Objectives:

- To learn about the Software architecture design and evaluation processes.
- To understand the concept of patterns and the Catalog.
- To know the Software architectures for product lines
- To understand the behavioral pattern and iterator pattern
- To understand design patterns to keep code quality high without overdesign.

UNIT I (9+3)


UNIT II (9+3)

Analyzing Architectures: Architecture evaluation, Architecture design decision making, Architecture Tradeoff Analysis Method, Cost-Benefit Analysis Method, Moving from one system to many: Software product lines, Building systems from off the shelf components, Software architecture in future.

UNIT III (9+3)

Patterns: Pattern description, Organizing catalogs, Role in solving design problems, Selection and usage, Creational and Structural Patterns: Abstract factory, Builder, Factory method, Prototype, Singleton, Adapter, Bridge, Composite, Façade, and Flyweight.

UNIT IV (9+3)

TEXT BOOKS:

REFERENCE BOOKS:

Course Learning Outcomes:
After completion of the course, the student will be able to

- design software architecture for large scale software systems
- describe a software architecture using various documentation approaches and architectural description languages
- identify and assess the quality attributes of a system at the architectural level
- communicate program structures using design patterns.
- select appropriate design patterns for design problems.
P14SE202 SOFTWARE QUALITY ASSURANCE AND TESTING

M.Tech. Semester: II Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Course Learning Objectives:

1. To understand the scope of software testing and quality assurance in software development life cycle
2. To know testing and quality assurance activities using modern software tools
3. To prepare test plans, schedules and budget for a testing and quality assurance projects
4. To manage testing and quality assurance projects

UNIT-I (9+3)

Software Quality: Perspective and expectations, Historical perspective of quality, Quality frameworks, Quality assurance as dealing with defects, Defect prevention detection and containment strategies, Quality Assurance Process and Quality Engineering: QA activities in software processes, Verification and validation perspectives, Reconciling the two views quality engineering, Activities and process quality planning, Goal setting and strategy formation quality assessment and improvement quality engineering in software processes.

UNIT-II (9+3)


UNIT-III (9+3)

Testing the Software with Dynamic White-Box Testing: Dynamic white-box testing, Dynamic white-box testing Vs. debugging, Testing the pieces, Data coverage, Code coverage, Configuration Testing: An overview of configuration testing, Approaching the task, Obtaining the hardware, Identifying hardware standards, Configuration testing other hardware, Compatibility Testing: Compatibility testing overview, Platform and application versions, Standards and guidelines data sharing compatibility, Usability Testing: User interface testing, What makes good user interface testing, Guidelines, Intuitive consistent, Flexible, Comfortable, Correct, Useful, Accessibility testing, Accessibility features in software.
UNIT-IV (9+3)


Text Books:


Reference Books:


Course Learning Outcomes:

After completion of the course, the student will be able to

- know the scope of software testing and quality assurance in software development life cycle
- capable of performing testing & quality assurance activities using modern software tools
- develop test plans, schedules and budget for a testing & quality assurance projects
- effectively manage a testing & quality assurance projects
P14SE203 ADVANCED DATA MINING

M.Tech. Semester: II Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>40 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>60 marks</td>
</tr>
</tbody>
</table>

Course Learning Objectives:
- To apply data mining techniques in real world applications
- To know advances in classification and clustering algorithms
- To develop web and text mining applications
- To know temporal and spatial mining applications

UNIT I (9+3)

Data mining overview: Mining frequent patterns, Associations and correlations, Classification and regression for predictive analysis, Cluster analysis, Outlier analysis, Pattern Mining Overview: Pattern mining in multilevel, Multidimensional space mining multilevel Associations, Mining multidimensional associations, Mining quantitative association rules, Mining rare patterns and negative patterns.

UNIT II (9+3)

Advance Classification: Classification by back propagation, Support vector machines, Classification using frequent patterns, Classification using rough sets and fuzzy sets. Advance Clustering: Density-based methods -DBSCAN, OPTICS, DENCLUE, Grid-Based methods - STING, CLIQUE, Exceptions: Maximization algorithm, Clustering Dimensional Data.

UNIT III (9+3)


UNIT IV (9+3)

Temporal Data Mining: Temporal association rules, Sequence mining, GSP algorithm, SPADE, SPIRIT episode discovery, Time series analysis, Spatial Mining: Spatial mining tasks, Spatial clustering, Data mining applications.
TEXT BOOKS:

REFERENCE BOOKS:

Course Learning Outcomes:
After completion of the course, the student will be able to

- apply the data mining algorithms for real world problems
- analyze advances in classification and clustering algorithms
- able to build web and text mining applications
- gain knowledge in temporal and spatial mining applications
P14SE 204    CLOUD COMPUTING

M.Tech. Semester: II    Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>40 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>60 marks</td>
</tr>
</tbody>
</table>

Course Learning Objectives:
- To introduce various cloud computing models.
- To understand cloud services and solutions.
- To know about cloud virtualization technologies and cloud management.
- To understand the relevance of cloud and SOA.

UNIT-I (9+3)


UNIT-II (9+3)

Cloud as a Service: Gamut of cloud solutions, Principal technologies, Cloud strategy, Cloud design and implementation using SOA, Conceptual cloud model, Cloud service defined, Cloud Solutions: Introduction, Cloud ecosystem, Cloud business process management, Cloud service management, Cloud stack, Computing on demand (CoD), Cloud sourcing.

UNIT-III (9+3)


UNIT-IV (9+3)

Text Books:

Reference Books:

Course Learning Outcomes:
After completion of the course, the student will be able to

- know the different cloud models.
- understand various services of cloud.
- gain knowledge on cloud virtualization technologies.
- learn cloud and SOA concepts
P14SE205A  MODEL DRIVEN SOFTWARE DEVELOPMENT

M.Tech. Semester: II  Specialization: Software Engineering

Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:

- To expose the students to the concepts of model driven software development
- To understand model driven software development relationships with other practices
- To manage projects based on model driven software development
- To know utilizing modern technologies for model driven software development

UNIT-I (9+3)

Model Driven Software Development (MDSD): basic ideas, terminology, challenges, goals, approaches and architecture, case study on web application, Model Concept formation:
Common model driven software development concepts and terminology, model driven architecture, architecture centric model driven software development, Generative programming.

UNIT-II (9+3)

MDSD Classifications: Model driven software development verses computer aided software engineering, 4GL, wizard, roundtrip engineering, Model driven software development and Patterns, Model driven software development and domain driven design, MDSD Capable Target Architecture: Software architecture in the context of MDSD, Building blocks of software architecture, Architecture reference model, balancing the MDSD platform, MDSD and component based development and Service oriented architecture.

UNIT-III (9+3)

Building domain architecture: Domain Specific Language construction, General transformation architecture, technical aspects of building transformations, and the use of interpreters, Code generation techniques: categorization, generation techniques Model transformations with Query view transformation, Model to model language requirements. MDSD tools, roles, architecture, selection criterion and pointers.

UNIT-IV (9+3)

Modular-based software design: Model-driven Architecture, Meta modeling, Meta levels vs Levels of abstraction, Model Driven Architectures Framework: Platform Independent Model, Platform Specific Model, System modeling- MOF's Meta modeling.
Text Book:

Reference Book:

Course Learning Outcomes:
After completion of the course, the student will be able to

- know essentials concepts model driven software development
- apply model driven software development for real time practices
- manage projects of model driven software development
- utilize the real-time technologies for model driven software development
P14SE205B  INFORMATION RETRIEVAL SYSTEM

M.Tech. Semester: II  Specialization: Software Engineering

Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation : | 40 marks |
| End Semester Exam : | 60 marks |

Course Learning Objectives:

- To expose the concepts of Information retrieval systems
- To analyze advances in information retrieval algorithms
- To know advances in web searching technologies
- To develop text classification based applications

UNIT I (9+3)

Boolean retrieval, Term vocabulary and postings lists, Dictionaries and tolerant retrieval, Index construction, Index compression.

UNIT II (9+3)

Scoring, term weighting and the vector space model, Computing scores in a complete search system, Evaluation in information retrieval, Relevance feedback and query expansion.

UNIT III (9+3)

XML retrieval, Probabilistic information retrieval, Language models for information retrieval, Web search basics, Web crawling and indexes, Link analysis.

UNIT IV (9+3)

Text classification, Vector space classification, Support vector machines and machine learning on documents, flat clustering, Hierarchical clustering, Matrix decompositions and latent semantic indexing.

TEXT BOOKS:

REFERENCE BOOKS:

Course Learning Outcomes:

After completion of the course, the student will be able to
- know essentials concepts Information retrieval systems
- analyze advances in information retrieval algorithms
- gain knowledge in advances of web searching technologies
- build text classification based application
P14SE205C  MACHINE LEARNING
M.Tech. Semester: II  Specialization: Software Engineering

Teaching Scheme :  

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation : | 40 marks |
| End Semester Exam : | 60 marks |

Course Learning Objectives:

- To understand the basic theory in machine learning.
- To understand a range of machine learning algorithms along with their strengths and weakness.
- To understand the machine learning algorithms implemented in different fields of computers.
- To formulate machine learning problems corresponding to different applications.
- To apply machine learning algorithms to solve problems of moderate complexity out code.
- To read current research papers and understands the issues raised by current research out code.

UNIT-I (9+3 Hrs)

Introduction: Learning, Types of machine learning, Supervised learning, Designing a learning system, Perspectives and issues in machine learning, Concept learning and the general to specific ordering – Introduction, A concept learning task, Concept learning as search, Find-S: finding a maximally specific hypothesis, Version spaces and the candidate elimination algorithm, Remarks on version spaces and candidate elimination, Inductive bias.

UNIT-II (9+3 Hrs)


UNIT-III (9+3 Hrs)

UNIT-IV (9+3 Hrs)


TEXT BOOKS:

REFERENCE BOOKS:

Course Learning Outcomes:
After completion of the course, the student will be able to
- know fundamental issues and challenges of machine learning: data, model selection, model complexity.
- know Strengths and weaknesses of many popular machine learning approaches.
- appreciate the underlying mathematical relationships within and across Machine Learning algorithms and the paradigms of supervised and un-supervised learning.
- design and implement various machine learning algorithms in a range of real-world applications.
P14 SE205D        SEMANTIC WEB AND SOCIAL NETWORKS

M.Tech.    Semester: II
Specialization: Software Engineering

Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

| Continuous Internal Evaluation : | 40 marks |
| End Semester Exam               : | 60 marks |

Course Learning Objectives:
- To learn Web Intelligence.
- To learn Ontology Engineering.
- To learn Knowledge Representation for the Semantic Web.
- To expose Semantic Web Applications, Services and Technology.
- To expose Social Network Analysis and semantic web.

UNIT-I (9+3)


UNIT-II: (9+3)

Ontology Engineering: Ontology engineering, Constructing ontology, Ontology development tools, Ontology methods, Ontology libraries and ontology mapping, Knowledge Representation for the Semantic Web: Ontologies and their role in the semantic web, Ontologies languages for the semantic web - resource description framework(RDF), RDF schema, Ontology web language(OWL), UML, XML/XML schema.

UNIT-III (9+3)

Semantic Web Applications, Services and Technology: Semantic web applications and services, Semantic search, e-learning, Semantic bioinformatics, Knowledge base, XML based web services, Creating an OWL-S ontology for web services, Semantic search technology, Web search agents and semantic methods.

UNIT-IV (9+3)

Text Books:


Reference Books:

Course Learning Outcomes:
After completion of the course, the student will be able to

- understand different techniques in web semantics.
- understand different tools, methods and mapping in Ontology engineering.
- analyze web services, semantic search techniques to develop semantic web applications.
- analyze social network structure and different sources for it.
P14 SE206A  BIG DATA ANALYTICS

M.Tech.  Semester: II  Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:

- Students will be able to learn about Big data analytic processes and tools.
- Students will be able to know about Big data architecture and reports.
- Students will be able to use Map reduce for building Big data applications.
- Students will be able to learn Frequent Item sets and Clustering.
- Students will be able to learn Frameworks and Visualization.

UNIT I (9+3)

Introduction: Velocity, Variety, Veracity, and Drivers for Big Data, Sophisticated consumers, Automation, Monetization, Big Data Analytics Applications: Social Media command center, Product knowledge hub, Infrastructure and operations studies, Product selection, Design and engineering, Location-based services, Online advertising, Risk management.

UNIT II (9+3)

Architecture Components: Massively parallel processing platforms, Unstructured Data Analytics and Reporting: Search and count, Context-sensitive and domain-specific searches, Categories and ontology, Qualitative comparisons, Data privacy protection, Real-Time adaptive analytics and decision engines, Advanced Analytics Platform: Real-Time architecture for conversations, Orchestration and synthesis using analytics engines, Entity resolution, Model management, Discovery using data at rest, Integration strategies.

UNIT III (9+3)

Implementation of Big Data Analytics: Revolutionary, Evolutionary or hybrid, Big Data governance, Integrating Big Data with MDM, Evolving maturity levels, Map-Reduce and New Software Stack: Distributed file systems, Physical organization of compute nodes, Large-scale file-system organization, Map-reduce features: Map tasks, Grouping by key, Reduce tasks, Combiners, Map-reduce execution, Coping with node failures, Algorithms using map-reduce for matrix multiplication, Relational algebra operations, Workflow systems, Recursive extensions to map-reduce.
UNIT IV (9+3)


TEXT BOOKS:

REFERENCES:

Course Learning Outcomes:
After completion of the course, the student will be able to

- learn about Big data analytic processes and tools.
- know about Big data architecture and reports.
- use Map reduce for building Big data applications.
- learn Frequent Item sets and Clustering.
- learn Frameworks and Visualization.
MOBILE COMPUTING

M.Tech. Semester: II
Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course learning objectives:
- To learn about the concepts and principles of mobile computing
- To distinguish between types of mobility
- To study the specifications and functionalities of various protocols of mobile networks
- To make students expertise in working with application protocols to develop mobile applications
- To explore both theoretical and practical issues of mobile computing

UNIT I (9+3)


UNIT II (9+3)


UNIT III (9+3)


UNIT IV (9+3)

TEXT BOOKS:

REFERENCE BOOKS:

Course Learning Outcomes:
After completion of the course, the student will be able to

- know the basic concepts and principles of mobile computing
- know the characteristics and limitations of mobile hardware devices including their user-interface modalities.
- understand the positioning techniques and location based services and applications
- know the structure and components for Mobile IP and Mobility management
- organize the functionalities and components of mobile computing systems into different layers.
P14SE206C  SOFT COMPUTING

M.Tech I Semester

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Specialization: Software Engineering

Examination Scheme:

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation:</th>
<th>40 Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam:</td>
<td>60 Marks</td>
</tr>
</tbody>
</table>

Course Learning Objectives:
- To learn the key aspects of soft computing.
- To know about the components and building block hypothesis of Genetic algorithm.
- To understand the features of neural network and its applications.
- To study the fuzzy logic components.
- To gain insight onto hybrid systems and Neuro Fuzzy modeling

UNIT-I (9+3)

UNIT-II (9+3)
Fuzzy Logic: Fuzzy sets, Operations on fuzzy sets, Fuzzy relations, Membership functions, Fuzzy rules and fuzzy reasoning, Fuzzy inference systems, Fuzzy expert systems, Fuzzy decision making

UNIT-III (9+3)

UNIT-IV (9+3)
Hybrid Systems: Integration of neural networks, Fuzzy logic, Genetic algorithms, Neuro fuzzy modeling, ANFIS architecture, CANFIS architecture, Classification and regression trees, Data clustering algorithms, Rule base structure identification.
Text Books:


Reference Books:


Course Learning Outcomes:

After the completion of course, the student will be able to

- implement machine learning through neural networks.
- gain knowledge to develop genetic algorithm.
- develop genetic algorithm to solve the optimization problem.
- develop a fuzzy expert system to derive decisions.
- model neuro fuzzy system for data clustering and classification.
P14SE206D DISTRIBUTED COMPUTING

M.Tech. Semester: II Specialization: Software Engineering

Teaching Scheme:  
<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>-</td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme:  
| Continuous Internal Evaluation | 40 marks |
| End Semester Exam | 60 marks |

Course Learning Objectives:
- To know the issues involved in distributed systems.
- To learn distributed computing paradigms.
- To learn computer networks architecture relevant to distributed computing.
- To understand basic knowledge of net-centric computing.

UNIT-I (9+3)

Introduction: Definition of distributed System, Characteristics of distributed systems, Goals of distributed system, Hardware concepts, Software concepts, Client-server model, Model of a distributed computation, Communication: Layered protocols, Remote procedure call, Remote object invocation, Message oriented communication, Stream oriented communication.

UNIT-II (9+3)


UNIT-III (9+3)


UNIT-IV (9+3)

Distributed Multimedia Systems: Characteristics of multimedia data, Quality of service management, Resource management, Stream Adaptation, Computing Technologies: Cluster/parallel computing, Coordination/Scheduling, Distributed object computing, Peer-to-peer computing, Service-oriented computing.
Text Books:

Reference Books:

Course Learning Outcomes:

After completion of the course, the student will be able to

- problem solving skills to distributed application.
- identify and decompose complex systems into its components parts
- integrate OS and programming language concepts to solve distributed components of the system.
- practice a variety of simple methods for develop suites of networking protocols for implementing the communicating components.
PI4SE 207 SOFTWARE TESTING LABORATORY

M.Tech. Semester: II  
Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Examination Scheme:

| Continuous Internal Evaluation | 40 marks |
| End Semester Exam               | 60 marks |

Course Learning Objectives:
To learn to use the following (or similar) automated testing tools to automate testing:
- Win Runner/QTP for functional testing.
- Load Runner for Load/Stress testing.
- Test Director for test management.
- JUnit, HTMLUnit, CPPUnit.

List of experiments on testing:
1. Write programs in ‘C’ Language to demonstrate the working of the following constructs:
   i) do...while ii) while....do iii) if...else iv) switch v) for
2. “A program written in ‘C’ language for Matrix Multiplication fails” Introspect the causes for its failure and write down the possible reasons for its failure.
3. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
4. Write the test cases for any known application (e.g. banking application)
5. Create a test plan document for any application (e.g. Library Management System)
6. Study of any testing tool (e.g. Win runner)
7. Study of any web testing tool (e.g. Selenium)
8. Study of any bug tracking tool (e.g. Bugzilla, bugbit)
9. Study of any test management tool (e.g. Test Director)
10. Study of any open source-testing tool (e.g. Test Link)
11. Take a mini project (e.g. University admission, Placement Portal) and execute it. During the Life cycle of the mini project create the various testing documents
12. Take a mini project (e.g. Library Management, Student register for a Course) and execute it. During the Life cycle of the mini project create the various testing documents.

Course Learning Outcomes:
After completion of the course, the student will be able to
- exposure on Win-runner and QTP for functional testing.
- use load runner for load and stress testing.
- use test director for test management.
- work with JUnit, HTMLUnit, CPPUnit.
Course Learning Objectives:

- To make students aware of real-time data warehousing tools
- To make students able to implement data mining algorithms
- To make students able to build data mining applications for credit risk analysis
- To make students capable to use WEKA tool for testing data mining algorithms

List of experiments:

1. Evolution of data management technologies, introduction to data warehousing concepts.
2. Develop an application to implement defining subject area, design of fact dimension table, data mart.
3. Develop an application to implement Extract, Transform and Load operations on a data warehouse.
4. Develop an application to implement OLAP, roll up, drill down, slice and dice operation.
5. Develop an application to construct a multidimensional data.
6. Develop an application to implement data generalization and summarization technique.
7. Develop an application to extract association rule of data mining.
8. Develop an application to extract data pattern.
9. Develop an application for classification of data.
10. Develop an application for decision tree.
11. Develop an application for clustering technique.
12. Develop an application to credit risk assessment.

Course Learning Outcomes:

After completion of the course, the student will be able to

- adopt real-time data warehousing tools
- implement data mining algorithms
- build data mining applications for credit risk management
- use WEKA tool for testing data mining algorithms
P14SE209 COMPREHENSIVE VIVA-VOCE

Class: M.Tech II Semester

Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>2</td>
</tr>
</tbody>
</table>

Examination Scheme:

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>--</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>100 marks</td>
</tr>
</tbody>
</table>

Guidelines:

There shall be only external oral examination for comprehensive viva-voce on a pre-notified date. The oral examination shall cover the entire content of courses covered in first and second semesters.
P14SE301     INDUSTRIAL TRAINING

M.Tech III Semester     Specialization: Software Engineering

Teaching Scheme :

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

Examination Scheme :

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>100 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>--</td>
</tr>
</tbody>
</table>

Guidelines for Industrial Training:

Coordinator in consultation with the Training & Placement Section has to procure training slots, for the students before the last day of instruction of 2\textsuperscript{nd} semester. The students shall confirm their training slots by the last day of 2\textsuperscript{nd} semester.

The students after 8 weeks Industrial Training shall submit a certificate, a report in the prescribed format before the last date specified by the Department Post Graduate Review Committee (DPGRC). The DPGRC shall evaluate their submitted reports and oral presentations.
P14SE302  DISSENGATION

Class : M.Tech III Semester  Specialization: Software Engineering

Teaching Scheme :

<table>
<thead>
<tr>
<th></th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
</tbody>
</table>

Examination Scheme :

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Weightage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissertation Supervisor Assessment</td>
<td>50%</td>
</tr>
<tr>
<td>DPGRC Assessment</td>
<td>50%</td>
</tr>
<tr>
<td><strong>Total Weightage:</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>

Guidelines for Dissertation:

Dissertation shall be normally conducted in two stages, spread over two sequential semesters i.e. third and fourth semester.

Registration Seminar shall be arranged within four weeks after completion of the Industrial Training and Seminar in the 3rd semester. The registration seminar shall include a brief report and presentation focusing the identified topic, literature review, time schedule indicating the main tasks, and expected outcome.

Progress Seminar-I: At the end of first stage (third semester), student shall be required to submit a preliminary report of work done for evaluation to the project coordinator and present the same before the DPGRC. The Continuous Internal Evaluation (CIE) for the third semester is as follows:
M.Tech IV Semester

Specialization: Software Engineering

Teaching Scheme:

<table>
<thead>
<tr>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>-</td>
<td>-</td>
<td>12</td>
</tr>
</tbody>
</table>

Examination Scheme:

<table>
<thead>
<tr>
<th>Continuous Internal Evaluation</th>
<th>100 marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Semester Exam</td>
<td>--</td>
</tr>
</tbody>
</table>

Guidelines for Dissertation:

Progress Seminar-II shall be arranged during the 6th week of IV semester.

Progress Seminar-III shall be arranged during the 15th week of IV semester.

Synopsis Seminar shall be arranged two weeks before the final thesis submission date. The student shall submit a synopsis report covering all the details of the works carried out duly signed by the dissertation supervisor.

At the end of second stage (fourth semester), student shall be required to submit two bound copies, one being for the department and other for the dissertation supervisor. The dissertation report shall be evaluated by the DPGRC and external examination shall be conducted on a pre-notified date. The dissertation evaluation for the fourth semester is as follows:

<table>
<thead>
<tr>
<th>Assessment</th>
<th>Weightage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dissertation Supervisor Assessment</td>
<td>20%</td>
</tr>
<tr>
<td>DPGRC Assessment</td>
<td>20%</td>
</tr>
<tr>
<td>ESE (Presentation &amp; Viva-voce)</td>
<td>60%</td>
</tr>
<tr>
<td><strong>Total Weightage:</strong></td>
<td><strong>100%</strong></td>
</tr>
</tbody>
</table>