# M.D.UNIVERSITY, ROHTAK
# SCHEME OF STUDIES AND EXAMINATION
# M.TECH 1st YEAR (SOFTWARE ENGINEERING)
# SEMESTER 1st
# CBCS Scheme effective from 2016-17

| **Sl. No** | **Course No.** | **Subject** | **Credit Pattern** | | | | **Examination Schedule (Marks)** | | | | **Duration of Exam (Hours)** | **No of Hours /week** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | **L** | **T** | **P** | **Total Credits** | **Marks of Class work** | **Theory** | **Practical** | **Total** | | |
| 1 | 16MSE21C1 | Advanced Software Engineering | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 2 | 16MSE21C2 | Software Project Management | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 3 | 16MSE21C3 | Software Architecture | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 4 | 16MSE21C4 | Web Development | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 5 | 16MSE21C5 | Analysis and Design of Algorithms | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 6 | 16MSE21C6 | Seminar | - | | - | 2 | 50 | - | - | 50 | | 2 |
| 7 | 16MSE21CL1 | Web Development Lab | - | - | 2 | 2 | 50 | - | 50 | 100 | 3 | 4 |
| 8 | 16MSE21CL2 | Analysis and Design of Algorithms Lab | - | - | 2 | 2 | 50 | - | 50 | 100 | 3 | 4 |
| | | **TOTAL** | | | | **26** | | | | | | |

NOTE:

**Examiner will set nine questions in total. Question One will be compulsory and will comprise short answer type questions from all sections and remaining eight questions to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each Unit.**

# M.D.UNIVERSITY, ROHTAK
## SCHEME OF STUDIES AND EXAMINATION
## M.TECH 1st YEAR (SOFTWARE ENGINEERING)
## SEMESTER 2nd
## CBCS Scheme effective from 2016-17

| Sl. No | Course No. | Subject | Credit Pattern | | | | Examination Schedule (Marks) | | | | Duration of Exam (Hours) | No of Hours/ week |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Total Credits | Marks of Class works | Theory | Practical | Total | | |
| 1 | 16MSE22C1 | Object Oriented Software Engineering | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 2 | 16MSE22C2 | Advanced Data Structures | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 3 | 16MSE22C3 | Seminar | - | | - | 2 | 50 | - | - | 50 | | 2 |
| 4 | 16MSE22CL1 | Advanced Data Structures Lab | - | - | 2 | 2 | 50 | - | 50 | 100 | 3 | 4 |
| 5 | 16MSE22CL2 | Operating Systems Lab | - | - | 2 | 2 | 50 | - | 50 | 100 | 3 | 4 |
| 6 | 16MSE22D1<br>Or<br>16MSE22D2<br>Or<br>16MSE22D3<br>Or<br>16MSE22D4 | Elective-1 | 4 | 0 | - | 4 | 50 | 100 | - | 150 | 3 | 4 |
| 7 | | Open Elective | 3 | 0 | - | 3 | | | | | | |
| 8 | | Foundation Elective | 2 | 0 | - | 2 | | | | | | |
| | | **TOTAL** | | | | **23** | | | | | | |

**NOTE: Examiner will set nine questions in total. Question One will be compulsory and will comprise short answer type questions from all sections and remaining eight questions to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each Unit.**

**Elective I: Choose any one from the following four papers:**

16MSE22D1 - Operating Systems
16MSE22D2 - Data Mining
16MSE22D3 - Software Reuse
16MSE22D4 -- Semantic Web

**Open Elective**: A candidate has to select this paper from the pool of Open Electives provided by the University.

**Foundation Elective:** A candidate has to select this paper from the pool of Foundation Electives provided by the University.

## Program Specific Outcomes (PSOs) – M.Techh(Software Engg)

At the end of the program, the student:

**PSO1**. Should be able to clearly understand the concepts and applications in the field of Software Engineering

**PSO2**. Should be able to associate the learning from the courses related to Databases, Operating Systems, Data Structures, Programming Languages to arrive at solutions to real world problems.

**PSO3**. Should have the capability to comprehend the technological advancements in the usage of modern tools to analyze and design subsystems/processes for a variety of applications.

**PSO4**. Should posses the skills to communicate in both oral and written forms, the work already done and the future plans with necessary road maps, demonstrating the practice of professional ethics and the concerns for societal and environmental wellbeing.

**PSO5**. Should be able to handle research problems and write dissertations.

# 16MSE21C1 ADVANCED SOFTWARE ENGINEERING

| | |
|---|---|
| L, T, P: 4, 0, 0 | Theory: 100 Marks |
| | Class work: 50 Marks |
| | Total: 150 Marks |
| | Duration of Exam: 3 Hours |

**Course Outcomes:-**

After completing this course, the student should be able to-

CO1 Have a higher-level understanding of how to write secure code by using the Secure Software Engineering Life Cycle.

CO2 Comprehend, apply, and implement Secure Software Requirements

CO3 Comprehend, apply, and implement Secure Software Design

CO4 Comprehend and apply Secure Software Implementation

CO5 Comprehend, apply, and implement Secure Software Testing

## UNIT 1

### INTRODUCTION:

System Concepts – Software Engineering Concepts - Software Life Cycle– Development Activities – Managing Software Development – Unified Modeling Language – Project Organization – Communication.

## UNIT 2

### ANALYSIS:

Requirements Elicitation – Use Cases – Unified Modeling Language, Tools – Analysis Object Model (Domain Model) – Analysis Dynamic Models – Non-functional requirements – Analysis Patterns.

## UNIT 3

### SYSTEM DESIGN, IMPLEMENTATION AND CHANGE MANAGEMENT:

Overview of System Design – Decomposing the system -System Design Concepts – System Design Activities – Addressing Design Goals – Managing System Design. Programming languages and coding- Human computer interaction-Reusing Pattern Solutions – Specifying Interfaces – Mapping Models to Code – Testing Rationale Management – Configuration Management – Project Management -real time interface design.

## UNIT 4

### ASPECT ORIENTED SOFTWARE DEVELOPMENT:

AO Design Principles -Separations of Concerns, Subject Oriented Decomposition, Traits, Aspect Oriented Decomposition, Theme Approach, Designing Base and Crosscutting Themes, Aspect-Oriented Programming using Aspect-J.

**References:**

1) Fundamentals of software Engineering, Rajib Mall, PHI
2) Software Engineering by Ian sommerville, Pearson Education.

3) Software Engineering – David Gustafson, 2002, Tata McGraw Hill.
4) Software Engineering Fundamentals, Oxford University, Ali Behforooz
5) An Integrated Approach to software engineering by Pankaj jalote.
6) Software Engineering by Roger S. Pressman, Tata McGraw Hill.

-----------------------------------------------------------------------------------------------------------

## 16MSE21C2 SOFTWARE PROJECT MANAGEMENT

| | |
|---|---|
| L, T, P: 4, 0, 0 | Theory: 100 Marks |
| | Class work: 50 Marks |
| | Total: 150 Marks |
| | Duration of Exam: 3 Hours |

**Course Outcomes:**

By the end of the course the students will be able to: -

CO1 Manage the scope, cost, timing, and quality of the project, at all times focused on project success as defined by project stakeholders.

CO2 Align the project to the organization's strategic plans and business justification throughout its lifecycle.

CO3 Identify project goals, constraints, deliverables, performance criteria, control needs, and resource requirements in consultation with stakeholders.

CO4 Interact with team and stakeholders in a professional manner, respecting differences, to ensure a collaborative project environment.

CO5 Adapt project management practices to meet the needs of stakeholders from multiple sectors of the economy Appraise the role of project management in organization change.

### UNIT 1

**PROJECT EVALUATION AND PROJECT PLANNING :** Importance of Software Project Management – Activities Methodologies – Categorization of Software Projects – Setting objectives – Management Principles – Management Control – Project portfolio Management – Cost-benefit evaluation technology – Risk evaluation – Strategic program Management – Stepwise Project Planning.

### UNIT 2

**PROJECT LIFE CYCLE AND EFFORT ESTIMATION :** Software process and Process Models – Choice of Process models – mental delivery– Rapid Application development – Agile methods – Extreme Programming – SCRUM – Managing interactive processes – Basics of Software estimation – Effort and Cost estimation techniques – COSMIC Full function points – COCOMO II A Parametric Productivity Model – Staffing Pattern.

### UNIT 3

**RISK MANAGEMENT:** Risk identification – Assessment – Monitoring – PERT technique - Monte Carlo simulation – Resource Allocation – Creation of critical patterns – Cost schedules.

**QUALITY PLANNING**: Quality Concepts, Procedural Approach to Quality Management, Quantitative Approaches to Quality Management, Quantitative Quality Management Planning, Setting the Quality Goal, Estimating Defects for Other Stages, Quality Process Planning, Defect Prevention Planning.

### UNIT 4

**PROJECT MANAGEMENT AND CONTROL:** Framework for Management and control – Collection of data – Visualizing progress – Cost monitoring – Earned Value

Analysis- Project tracking – Change control- Software Configuration Management – Managing contracts – Contract Management.

**STAFFING IN SOFTWARE PROJECTS :** Managing people – Organizational behavior – Best methods of staff selection – Motivation – The Oldham-Hack man job characteristic model - Ethical and Programmed concerns – Working in teams – Decision making – Team structures - Virtual teams – Communications genres – Communication plans.

**TEXTBOOK:**

1) Bob Hughes, Mike Cotterell and Rajib Mall: Software Project Management – Fifth Edition, Tata McGraw Hill, New Delhi, 2012.

**REFERENCES:**

1) Robert K. Wysocki "Effective Software Project Management" Wiley Publication,2011.
2) Pankaj Jalote, Software project management in practice, Addison-Wesley
3) Walker Royce: "Software Project Management"- Addison-Wesley, 1998.
4) Gopalaswamy Ramesh, "Managing Global Software Projects" – McGraw Hill Education (India), Fourteenth Reprint 2013.

-----------------------------------------------------------------------------------------------------------

## 16MSE21C3 SOFTWARE ARCHITECTURE

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcomes-**

As a result of completing this course, student will be able to

CO1 Understand architectures in Agile projects

CO2 Define what a software architecture is and explain why it is important

CO3 Understand software architectural patterns and tactics and their relationship to system qualities

CO4 Appreciate the importance of documenting software architecture

CO5 Explain methods for evaluating software architecture

### UNIT 1

**Software Architecture Terms:** Component, Relationship, View, Architectural styles, Frameworks, Methodologies, Processes, Functional and Non-functional properties of software architecture.

### UNIT 2

**Enabling Techniques for Software Architecture:** Abstraction, encapsulation, Information Hiding modularization, Separation of concerns, coupling and Cohesion. Sufficiency, Completeness and Primitiveness separation of policy and implementation. Separation of interface and implementation

### UNIT 3

**Architectural Styles:** pipes and filters, Data abstraction and object-orientation, Event-based, implicit invocation, Layered systems, Repositories, Interpreters, Process control, Heterogeneous Architectures.

### UNIT 4

**Software Implementation-Development Environment Facilities:** code generation, reverse engineering, profiling, software libraries, testing and debugging. Software quality: changeability, efficiency, interoperability, Reliability, testability, reusability, fault tolerant software.

**References:**

1) M.Shaw: Software architecture perspective on an Emerging Discipline, Prentice Hall .
2) Len Bass. Paul clements, Rick Kazman : software architecture in practice, pearson education Asia.

---

## 16MSE21C4 WEB DEVELOPMENT

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcome:**

By the end of course the student will be able to:

CO1 Understand java concepts and applicability contexts.
CO2 Learn about HTML, PERL, VB Script and Java Script.
CO3 Learn server side technologies like CGI, ASP and JSP.

### UNIT 1

**Web search basics:** Background and history, Web characteristics, Advertising as the economic model ,The search user experience, Index size and estimation, Near-duplicates and shingling.

### UNIT 2

**Web crawling and indexes:** Overview, Crawling, Distributing indexes, Connectivity servers. Link analysis: The Web as a graph, PageRank, Hubs and authorities

### UNIT 3

**Web Servers (IIS/PWS & Apache):** HTTP request types, system architecture, client-side scripting, accessing web servers requesting documents, HTTP, secure HTTP, Secure Sockets Layer, WWW Proxies, Basic Feature Bookmarks, Cookies, Progress Indicators, Customization of Browsers, Browsing Tricks, Next Generation Web Browsing,

### UNIT 4

**DHTML, XHTML, AJAX, XML:** Structuring data, XML namespaces, DTD and schemas, XML variables, DOM methods, simple API for XML, Web services, and application of XML. Active Server Pages (ASP): How ASP works, ASP objects, file system, objects, ASP.NET

**References:**

1) Fundamentals of the Internet and the World Wide Web, Raymond GreenLaw and Ellen Hepp-2011, TMH.
2) Internet and World Wide Web Programming, Deitel, Deitel and Neito, 2000, Pearson
3) Introduction to Information Retrieval, Christopher D. Manning, Prabhakar Raghavan & Hinrich Schutze, Cambridge university press, 2008
4) Beginning XHTML by Frank Boumpery, Cassandra Greer, Dave Ragett, Jenny Ragett, Subastia Schintenbaumer and Ted Wugofski 2000,WROX Press(Indian Shroff Publication SPD)1st Edison.
5) Complete Reference Guide to Java Script, Aron Weiss,QUIE,1977.
6) Intranet and Internet Engg. By Minoli.

## 16MSE21C5 ANALYSIS & DESIGN OF ALGORITHMS

L, T, P: 4, 0, 0

Theory: 100 Marks<br>
Class work: 50 Marks<br>
Total: 150 Marks<br>
Duration of Exam: 3 Hours

**Course Outcomes**:

By the end of course the student will be able to:

CO1 Analyzing worst case analysts of algorithms on the basis of asymptotic notations.

CO2 Comparing the different data structures and choosing the best one according to the problem.

CO3 Describe the greedy, divide and conquer and dynamic programming approach of the problem solving and deciding when to use the particular approach.

CO4 Prove the correctness and analyze the running time of the basic algorithms for those classic problems in various domains.

### UNIT 1

**ANALYSIS ALGORITHMS & PROBLEMS:** Introduction to algorithms, Time and Space Complexity, Basic elements of data structures like link lists, Stacks and Queues, Trees , Graphs, Recursion. Different types of sorting algorithms and their complexities. DYNAMIC SETS, SEARCHING GRAPHS : Introduction, Array, amortized time analysis, red black trees, hashing heaps, dynamic equivalence relations and union-find programs, priority queues with decrease key operations, traversing graphs, DFS strongly connected components, bio connected components, minimum spanning tree algo., single source shortest paths, all pair shortest paths.

### UNIT 2

**GREEDY AND DYNAMIC METHODS:** Introduction to greedy and dynamic methods, their algorithms and comparative study.

### UNIT 3

**BACKTRACKING AND BRANCH & BOUND:** General backtracking and branch and Bound methods, 8 queen, sum of subset, graph coloring, Hamilton cycles, 0/1 knapsack problem.

### UNIT 4

**NP HARD AND NP COMPLETE PROBLEMS:** Basic Concepts, Cooks Theorem, P – Hard graph problems, NP Hard Scheduling.

**References:**

1) Computer Algorithms: Introduction to design and analysis (3rd edition ) by Sara Baase and Allen Van Gelder , Pearson, 2000.
2) Fundamentals of Algorithms by Gilles Brassard and Paul Bratley

3) Design and Analysis of Algorithms ( Computer science Series) by Jeffrey D. Smith Publ.
4) Fundamentals of Computer algorithms, Ellis Horowitz and SratajSahnim 1978, Galgotia publ.
5) Algorithms Design (PIE) by Eva Tardos and Jon Klienberg, person.
6) Introduction to Algorithms, Thomas h Cormen, Harles E leiserson and Ronald Lrivest
: 1990, TMH.

---

## 16MSE21C6 SEMINAR

L, T, P: 0, 0, 2 Class work: 50 Marks

Total: 50 Marks

**Course Outcomes**

At the end of the course the student will be able to:

CO1 Ability to analyze the selected topic, organize the content and communicate to audience in an effective manner

CO2 Ability to practice the learning by self study a candidate has to present a seminar on a recent topic/ technology/ research advancement

CO3 make a seminar report.

## 16MSE21CL1 WEB DEVELOPMENT LAB

| | |
|---|---|
| L, T, P: 0, 0, 3 | Theory: 50 Marks |
| | Class work: 50 Marks |
| | Total: 100 Marks |

**Course Outcomes:**

At the end of the course, students should be able to:
CO1 Design and implement dynamic websites with good aesthetic sense of designing and latest technical know-how's.
CO2 Have a Good grounding of Web Application Terminologies, Internet Tools, E – Commerce and other web services.
CO3 Get introduced in the area of Online Game programming.

**Suggestive list of experiments**:

1) WEBBASICS: Design web pages through coding using HTML and DHTML.
2) Integrated Development Tool: Frontpage2000/Dreamweaver
3) BROWSER SIDE SCRIPTING using JavaScript with a focus on
4) Event Handling and Validation
5) SERVER SIDE SCRIPTING:
6) PHP SYNTAX, variables, loops and constructs.
7) JAVA GRAPHICS
8) BROWSER SIDE SCRIPTING: Introduction to programming world of XML Technologies. Basic XML Tags, Database Handling with PHP and XML. Connecting to Databases using PHP, PHP files and databases. Advanced XML: XLINK, XQUERY, XPATH, AppML (XML Language for Internet Application), SCHEMA, DTD, DOM, RDF, RSS. AJAX (Asynchronous JavaScript and XML), E4X (New Extension to JavaScript, its Direct support of XML to JavaScript)
9) JAVA GAME PROGRAMMING AND XML (XSL, XSLT): 2D, 3D Graphics, Event Handling and Developing Online Games.

-----------------------------------------------------------------------------------------------------------

## 16MSE21CL2 ANALYSIS AND DESIGN OF ALGORITHMS LAB

| | |
|---|---|
| L, T, P: 0, 0, 3 | Theory: 50 Marks |
| | Class work: 50 Marks |
| | Total: 100 Marks |

**Course Outcome:**

After completion of this course student will be able to:-
CO1 Identify Data Structures, Design paradigms and Computational complexity in the design of simple tools

CO2 Demonstrate relationships among NP-Complete Problems

CO3 Implement the approximate algorithms approach to solve some NP-Complete Problems. Demonstrate randomness by solving some examples

CO4 Implement algorithms for geometry and large data-sets.

**Suggestive list of experiments**:-

1) Design of simple tool of choice for revising of basic concepts.
2) Implement program to show relationship between clique, vertex-cover and independent set.
3) Implement approximate algorithm technique for vertex cover/ steiner tree.
4) Implement approximate algorithms on various variants of Travelling Salesman problem.
5) Demonstrate randomness by implementing Quick sort algorithm.
6) Demonstrate randomness by implementing min-cut algorithm.
7) Demonstrate with a Program the Markov property and stationery markov chain
8) Implementing the Viterbi algorithm
9) Implementing the forward algorithm

Implementing algorithms from geometry problems and large data sets

-----------------------------------------------------------------------------------------------------------

## 16MSE22C1 OBJECT-ORIENTED SOFTWARE ENGINEERING

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcome:**

By the end of course the student will be able to:
CO1 Employ formal methods to produce effective software designs as solutions to specific tasks.
CO2 Develop structured sets of simple user-defined classes using Object-Oriented principles
CO3 learn UML for OO design

### UNIT 1

**Introduction –** Overview of Object-Orientation; Basic Concepts of Object-Orientation: Data abstraction, Encapsulation, Inheritance, Aggregation, Classes, Objects, Messages, Inheritance, Polymorphism, Object oriented modeling, OO Life cycle models.

### UNIT 2

**Object Oriented Methodology:** Rumbaugh methodology, Booch methodology, Jacobson methodology, Patterns, frameworks, the unified modeling language (UML).

**Requirement Elicitation:** Concept, Activity, Techniques, Requirements Model-Action & Use cases.

**Architecture:** Requirement Model, Analysis Model, Design Model, Implementation Model, Test Model.

### UNIT 3

**Object–Oriented Analysis:** Use-Case Driven Object Oriented Analysis, Use-Case Model, Object Classification, Classification Theory, Approaches for identifying classes, classes, responsibilities and collaborators, identifying Object Relationships, attributes and Methods. Object –Oriented Design process and design Axioms

### UNIT 4

**Testing Object System:** Introduction, Testing Activities and Techniques, Testing processes, managing testing.

**Text Books:**

1) Ivar Jacobson, “Object Oriented Software Engineering”, Pearson.
2) Grady Booch, James Runbaugh, Ivar Jacobson, “The UML User Guide”, Pearson.

**Reference Books:**

3) Rumbaugh et. al, “Object Oriented Modeling and Design”, Pearson.
4) Booch, Maksimchuk, Engle, Young, Conallen and Houstan, “Object Oriented Analysis and Design with Applications”, Pearson Education.

----------------------------------------------------------------------------------------------------

## 16MSE22C2 ADVANCED DATA STRUCTURE

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcomes-**

CO1 Student will be able to choose appropriate data structure as applied to specified problem definition.

CO2 Student will be able to handle operations like searching, insertion, deletion, traversing mechanism etc. on various data structures.

CO3 Students will be able to apply concepts learned in various domains like DBMS, compiler construction etc.

CO4 Students will be able to use linear and non-linear data structures

### UNIT 1

**Data Structure:** Introduction to data structures and algorithms, Linear and nonlinear data structures.

**Algorithm Analysis:** time complexity and space complexity, Time space trade off and Big O Notation, efficiency of algorithms, Worst case and average case analysis.

**Arrays:** Introduction, one dimensional and multidimensional arrays, memory representation of arrays, operations on arrays, sparse matrices and their implementation, Advantages and limitation of arrays.

**Stacks:** Introduction; Operation on stacks, Static and Dynamic Implementation of stacks **Application of Stacks:** matching parenthesis, evaluation of arithmetic expressions, conversion from infix to postfix, recursion.

**Queues:** introduction, operation on queues, Static and Dynamic Implementation of Queues, circular queues, Dequeues, priority queues, application of queues.

### UNIT 2

**Linked List:** Introduction, Operations on linked list, Circular linked list, Doubly linked list, Header linked list, Implementation of linked list, Application of linked lists.

### UNIT 3

**Trees:** Introduction, Binary Tree, Tree traversal Algorithms, Threaded Binary Trees, Binary Search Tree, AVL Tree, M-way search tree, B-Trees, Heap.

**Graphs:** Introduction, Memory Representation of Graphs: adjacency matrix representation of graphs, adjacency list or linked list representation of graphs. Operations performed on graphs, Graph traversal algorithms, Shortest Path algorithm, Minimum Spanning Tree, Applications of graph.

### UNIT 4

**Sorting and Searching:** Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Radix Sort, Quick Sort and Heap Sort. Linear Search, Binary Search.

Hashing - Hashing Functions, Collision Resolution Techniques.

**References:**

1) Tanenbaum, A. S., “Data Structures using ‘C’”, PHI
2) Seymour Lipschultz, “Theory and Practice of Data Structures", McGraw-Hill
3) S. Sahni, "Data Structures, Algorithms and Application in C++, McGraw-Hill
4) M. A. Weiss, "Data Structures and Algorithm Analysis in C++, Pearson Education,
5) Aho Alfred V., Hopperoft John E., UIlman Jeffrey D., “Data Structures and Algorithms”, Addison Wesley
6) Drozdek- Data Structures and Algorithms.

---

## 16MSE22D1 OPERATING SYSTEMS

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcomes-**

CO1 Demonstrate understanding of the concepts, structure and design of operating Systems

CO2 Demonstrate understanding of operating system design and its impact on application

CO3 Describe, contrast and compare differing structures for operating systems.

CO4 understand and analyze theory and implementation of: processes, resource control.

CO5 Understand the high-level structure of the Linux kernel both in concept and source code

### UNIT 1

**Fundamentals of Operating system**: Overview – Synchronization Mechanisms – Processes and Threads - Process Scheduling – Deadlocks: Detection, Prevention and Recovery – Models of Resources – Memory Management Techniques.

### UNIT 2

**Distributed Operating Systems:** Issues in Distributed Operating System – Architecture – Communication Primitives – Lamport's Logical clocks – Causal Ordering of Messages – Distributed Mutual Exclusion Algorithms – Centralized and Distributed Deadlock Detection Algorithms – Agreement Protocols.

### UNIT 3

**Distributed Resource Management:** Distributed File Systems – Design Issues - Distributed Shared Memory – Algorithms for Implementing Distributed Shared memory–Issues in Load Distributing – Scheduling Algorithms – Synchronous and Asynchronous Check Pointing and Recovery – Fault Tolerance – Two-Phase Commit Protocol – Nonblocking Commit Protocol – Security and Protection.

### UNIT 4

**Real Time And Mobile Operating Systems** : Basic Model of Real Time Systems - Characteristics- Applications of Real Time Systems – Real Time Task Scheduling - Handling Resource Sharing - Mobile Operating Systems –Micro Kernel Design - Client Server Resource Access – Processes and Threads - Memory Management - File system. CASE STUDIES of Linux System: Design Principles - Kernel Modules - Process Management Scheduling - Memory Management - Input-Output Management - File System – Inter-process Communication. iOS and Android: Architecture and SDK Framework - Media Layer - Services Layer - Core OS Layer

File System.

**References:**

1) Mukesh Singhal and N. G. Shivaratri, “Advanced Concepts in Operating Systems”, McGrawHill, 2000
2) Abraham Silberschatz, Peter B. Galvin, G. Gagne, “Operating System Concepts”, Sixth Addison n Wesley Publishing Co., 2003.
3) Andrew S. Tanenbaum, “Modern Operating Systems”, Second Edition, Addison Wesley, 2001.

---

## 16MSE22C3 SEMINAR

L, T, P: 0, 0, 2

Class work: 50 Marks
Total: 50 Marks

**Course Outcomes:-**

At the end of the course the student will have:

CO1 Ability to analyze the selected topic, organize the content and communicate to audience in an effective manner

CO2 Ability to practice the learning by self study

**-**

## 16MSE22CL1 ADVANCE DATA STRUCTURE LAB

L, T, P: 0, 0, 3

Class work: 50 Marks
Sessional: 50 Marks
Total: 100 Marks

**Course Outcomes:-**

By the end of this course the students

CO1 Be familiar with basic data structure of algorithms.

CO2 Be familiar with writing recursive methods by using C++.

CO3 Master the implementation of linked data structures such as linked lists and binary trees.

CO4 Be familiar with advanced data structures such as balanced search trees, hash tables, priority queues and the disjoint set union/find data structure

CO5 Be familiar with several searching and sorting algorithms including quicksort, Merge-Sort and Heap-Sort.

## 16MSE22CL2 OPERATING SYSTEMS LAB

L, T, P: 0, 0, 3

Class work: 50 Marks
Sessional: 50 Marks
Total: 100 Marks

**Course Outcomes-**

By the end of this course the students

CO1 Will demonstrate understanding of operating system design and its impact on application

CO2 Analyze and simulate CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority

CO3 Implement memory management schemes and page replacement schemes.

CO4 Simulate file allocation and organization techniques.

## 16MSE22D2 DATA MINING

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcomes:**

By the end of this course the student will have

CO1 Ability to understand social media and its data.

CO2 Ability to apply mining technologies on twitter, facebook, LinkdIn and Googlet.

CO3 Ability to apply web mining technologies, NLP concepts to summarize, mine data on webpage, blogs.

CO4 Ability to apply search engines

### UNIT 1

**Data Mining:** Introduction, association rules mining, Naive algorithm, Apriori algorithm, direct hashing and pruning (DHP), Dynamic Item set counting (DIC), Mining frequent pattern without candidate generation(FP, growth), performance evaluation of algorithms, Mining Customer values: From Association rule to direct mining: A case study.

### UNIT 2

**Classification:** Introduction, decision tree, tree induction algorithm – split algorithm based on information theory, split algorithm based on Gini index; naïve Bayes method; estimating predictive accuracy of classification method; classification software, software for association rule mining; case study; KDD Insurance Risk Assessment: A Case study.

### UNIT 3

**Cluster analysis:** Introduction, partitional methods, hierarchical methods, density based methods, dealing with large databases, cluster software; Efficient Clustering of Very Large Document Collections: A case study.

### UNIT 4

**Web Data Mining:** Web Terminology and Characterstics, Locality and Hierarchy in the web, Web Content Mining, Web Usage Mining, Web Structure Mining, Web mining Software. Search engines: Characterstics of Search engines, Search Engine Functionality, Search Engine Architecture, Ranking of web pages, the search engine history, Enterprise Search, Enterprise Search Engine Software.

**References :**

1) Han J., Kamber M. and Pei J., Data mining concepts and techniques, Morgan Kaufmann Publishers (2011) 3rd ed.
2) Pudi V., Krishana P.R., Data Mining, Oxford University press, (2009) 1st ed.
3) Adriaans P., Zantinge D., Data mining, Pearson education press (1996), 1st ed.
4) Pooniah P., Data Warehousing Fundamentals,Willeyinterscience Publication, (2001)

---------------------------------------------------------------------------------------------------------

## 16MSE22D3 SOFTWARE REUSE

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcomes**

At the end of this course, student would be able to

CO1 Get the foundations and concepts of software reuse

CO2 Advocate the importance and means of reusability

CO3. Understand the basic concepts of object oriented domain engineering

### UNIT 1

**Introduction Software Reuse and Software Engineering**, Concepts and Terms, Software Reuse Products, software reuse Processes, Software Reuse Paradigms.

**State of the art and the Practice:** Software reuse management, Software reuse techniques, Aspects of software reuse, Organisational aspects, Technical aspects and Economical aspects.

### UNIT 2

**Programming Paradigms and Reusability:** Usability Attributes, Representation and Modelling Paradigms, Abstraction and Composition in development paradigm.

### UNIT 3

**Object-Oriented DomainEngineering:** Abstraction and Parameterized techniques, Composition techniques in Object Orientation.

### UNIT 4

**Application Engineering:** Component Storage and Retrieval, Reusable Asset Integration. Software Reuse Technologies: Component based Software Engineering, COTs based development, Software Reuse Metrics, Tools for Reusability.

**References:**

1) Reuse Based Software techniques, Organisation and Measurement by Hafedh Mili Sherif Yacoub and Edward Eddy, John wiley &Sons Inc.
2) The Three Rs. Of Software Automation: Re-engineering, Repository, Reusablity by Cama MCClure, Prentice Hall.
3) McClure,Carma,L.Software Reuse Techniques: adding reuse to the system development processes/:Prentice Hall.
4) Poulin, Jeffery S. Measuring software reuse : principles, practices and economic models/ Jeffery S.Poulin Reading, Mass:Addison Wesley.

---------------------------------------------------------------------------------------------------------------

## 16MSE22D4 SEMANTIC WEB

L, T, P: 4, 0, 0

Theory: 100 Marks
Class work: 50 Marks
Total: 150 Marks
Duration of Exam: 3 Hours

**Course Outcomes:**

By the end of this course the student will

CO1 Understand concepts of semantic web.
CO2 Understand and carry out proper experimental work for ontologies
CO3 be able to design, implement, and analyze XML
CO4 Able to understand OWL

### UNIT 1

**Introduction to the Syntactic web and Semantic Web** – Evolution of the Web – The visual and syntactic web – Levels of Semantics – Metadata for web information - The semantic web architecture and technologies –Contrasting Semantic with Conventional Technologies – Semantic Modeling - Potential of semantic web solutions and challenges of adoption

### UNIT 2

**Ontologies – Taxonomies –Topic Maps – Classifying Ontologies –** Terminological aspects: concepts, terms, relations between them – Complex Objects –Subclasses and Sub-properties definitions – Upper Ontologies – Quality – Uses - Types of terminological resources for ontology building – Methods and methodologies for building ontologies – Multilingual Ontologies -Ontology Development process and Life cycle – Methods for Ontology Learning – Ontology Evolution – Versioning

### UNIT 3

**Structured Web Documents - XML** – Structuring – Namespaces – Addressing – Querying – Processing - RDF – RDF Data Model – Serialization Formats- RDF Vocabulary –Inferencing - RDFS – basic Idea – Classes – Properties- Utility Properties – RDFS Modeling for Combinations and Patterns- Transitivity

### UNIT 4

**OWL – Sub-Languages – Basic Notions** -Classes- Defining and Using Properties – Domain and Range – Describing Properties - Data Types – Counting and Sets- Negative Property Assertions

**Advanced Class Description** – Equivalence – Owl Logic. Development Tools for Semantic Web – Jena, Protégé, Swoop and TopBraid Composer – SPARQL –Querying semantic web - Semantic Wikis - Semantic Web Services – Modeling and aggregating social network data -

Ontological representation of social relationships, Aggregating and reasoning with social network data

**References:**

1) Rajendra Akerkar: “Foundations of the Semantic Web”, Narosa Publishing House, New Delhi and Alpha Science Intern., Oxford 2009
2) Jeffrey T. Pollock: “Semantic Web for Dummies”, John Wiley, 2009.
3) Liyang Yu, “A Developer's Guide to the Semantic Web”, Springer, First Edition, 2011
4) John Hebeler, Matthew Fisher, Ryan Blace and Andrew Perez-Lopez, “Semantic Web Programming”, Wiley, First Edition, 2009.
5) Grigoris Antoniou, Frank van Harmelen, “A Semantic Web Primer”, Second Edition (Cooperative Information Systems) (Hardcover), MIT Press, 2008
6) Robert M. Colomb, “Ontology and the Semantic Web”, Volume 156 Frontiers in Artificial Intelligence and Applications (Frontier in Artificial Intelligence and Applications), IOS Press, 2007.
7) Dean Allemang and James Hendler, “Semantic Web for the Working Ontologist: Effective Modeling in RDFS and OWL, Morgan Kaufmann”, Second Edition, 2011.
8) Thomas B. Passin, “Explorer’s Guide to the Semantic Web (Paperback)”, Manning Publications 8 Jul 2004

**M.D UNIVERSITY**

**SCHEME OF STUDIES AND EXAMINATION M.TECH 2nd YEAR (SOFTWARE ENGINEERING) SEMESTER 3rd**
**CBCS Scheme effective from 2017-18**

| Sr. No | Course No. | Subject | Teaching Schedule | | | | Examination Schedule (Marks) | | | | Duration of Exam (Hours) | No of hours/ week |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Total credits | Marks of Class works | Theory | Practical | Total | | |
| 1 | 17MSE23C1 | Software Testing | 4 | 0 | - | 4 | 50 | 100 | - | 150 | **3** | **4** |
| 2 | 17MSE23C2 | Advance Database Management System | 4 | 0 | - | 4 | 50 | 100 | - | 150 | **3** | **4** |
| 3 | 17MSE23C3 | Literature Survey (Dissertation Stage 1) | - | - | 2 | 2 | 100 | - | - | 100 | | **4** |
| 4 | 17MSE23C4 | Seminar | - | | 2 | 2 | 50 | - | - | 50 | | **2** |
| 5 | 17MSE23CL1 | Advance Database Management System Lab | - | - | 2 | 2 | 50 | - | 50 | 100 | | **2** |
| 6 | 17MSE23CL2 | Project | - | - | 2 | 2 | 50 | - | 50 | 100 | | **2** |
| 7 | | Foundation Elective | | | | 2 | | | | | | |
| 8 | | Open Elective | | | | 3 | | | | | | |
| | | **TOTAL** | | | | **21** | | | | | | |

## NOTE:

Examiner will set nine questions in total. Question One will be compulsory and will comprise of all sections and remaining eight questions to be set by taking two questions from each unit. The students have to attempt five questions in total, first being compulsory and selecting one from each Unit.

## Foundation Elective

A candidate has to select this paper from the pool of Foundation Electives provided by the University.

## OPEN ELECTIVE

A candidate has to select this paper from the pool of open electives provided by the

University.

## M.D. UNIVERSITY

## SCHEME OF STUDIES AND EXAMINATION M.TECH 2nd YEAR (SOFTWARE ENGINEERING) SEMESTER 4th
## CBCS Scheme effective from 2017-18

| Sr. No | Course No. | Subject | Teaching Schedule | | | | Examination Schedule (Marks) | | | | No of Credits |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L | T | P | Total | Marks of Class works | Theory | Practical | Total | |
| 1. | 17MSE24C1 | Dissertation and viva (Dissertation Stage 2) | - | - | - | - | 250 | - | 500 | 750 | **20** |
| | | **TOTAL** | **-** | **-** | **-** | **-** | | | | | |

NOTE:

1. **Students have to publish a research paper in a journal / conference of the research work done in the semester.**

# 17MSE23C1-SOFTWARE TESTING

| | |
|---|---|
| L, T, P: 4, 0, 0 | Theory: 100 Marks |
| | Class work: 50 Marks |
| | Total: 150 Marks |
| | Duration of Exam: 3 Hours |

**Course Outcomes**

By the end of this course the student will

CO1 Know the basic concepts of software testing and its essentials.

CO2 Able to identify the various bugs and correcting them after knowing the consequences of the bug.

CO3 Use of program's control flow as a structural model is the corner stone of testing.

CO4 Performing functional testing using control flow and transaction flow graphs.

CO5 Know the basic techniques for deriving test cases

## CO1 - UNIT I

**Testing as an engineering activity –** Role of process in software quality – Testing as a process – Basic definitions – Software testing principles – The tester's role in a software development organization – Origins of defects – Defect classes – The defect repository and test design – Defect examples – Developer / Tester support for developing a defect repository.

## UNIT II

**Introduction to testing design strategies –** The smarter tester – Test case design strategies – Using black box approach to test case design – Random testing – Equivalence class partitioning – Boundary value analysis – Other black box test design approaches – Black box testing and COTS – Using white box approach to test design – 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 – Evaluating test adequacy criteria.

## UNIT III

**The need for levels of testing –** Unit test – Unit test planning – Designing the unit tests – The class as a testable unit – The test harness – Running the unit tests and recording results – Integration tests – Designing integration tests – Integration test planning – System test – The different types – Regression testing – Alpha, beta and acceptance tests.

## UNIT IV

**Basic concepts –** Testing, debugging goals, policies – Test planning – Test plan components – Test plan attachments – Locating test items – Reporting test results – The role of three groups in test planning and policy development – Process and the engineering disciplines – Introducing the test specialist – Skills needed by a test specialist – Building a testing group. Defining terms – Measurements and milestones for controlling and monitoring – Status meetings – Reports and control issues – Criteria for test completion – Types of reviews – Developing a review program – Components of review plans – Reporting review results.

## References:

1) Foundation of Software Testing by Aditya P. Mathur, Pearson Publication.
2) Software Testing, A Craftsman Approach by Paul. C. Jorgensen, CRC Press.
3) Software Testing- Effective Methods, Tools and Techniques by Renu Rajani, Pradeep Oak, Tata McGraw hill Publications.
4) Software Testing: Principles and Practices by Srinivasan Desikan and Ramesh, Pearson Education.
5) The Art of Software Testing by Glenford J. Myers, Wiley Publications.

-----------------------------------------------------------------------------------------------------------

## 17MSE23C2-ADVANCE DATABASE MANAGEMENT SYSTEM

| | |
|---|---|
| L, T, P: 4, 0, 0 | Theory: 100 Marks |
| | Class Work: 50 Marks |
| | Total: 150 Marks |
| | Duration of Exam: 3 Hours |

**Course Outcomes:**

At the end of this course the student will be able to

CO1 Explain the basic concepts and the applications of database systems.

CO2 Utilize the knowledge of basics of SQL and construct queries using SQL.

CO3 Use commercial relational database system (Oracle) by writing Queries using SQL.

CO4 Apply relational database theory, and be able to write relational algebra expressions for queries.

## UNIT I

**Introduction: -**Traditional Approach to Information processing. Data Base Concepts and its Approach to data processing. Feature of DBMS, Software's, Users, why Database? What is BDBMS. Elements of DBMS: Data definition languages(DDL). Data Manipulation Languages (DML). Data Query Language (DQL). How does a DBMS works?

## UNIT II

**Introduction to Data base model**, relational data base models, hierarchical database, network database, database design, applications, problems in DBMS environment, selecting database software, basic of relational database management.

Relational algebra (union, intersection, difference, Cartesian product, select, project, join, divide), entity-relationship model, components (entities, attributes, relationship, cardinality, weak entities, recursive entities).

## UNIT-III

**Normalization**, the need for the normalization, conversion to Ist, IInd and IIIrd normal form, file access method-sequential file, direct access, role of DBMS, why relational database, structure of dbms, next generation data base system, knowledge based system, computer facilities, features of distributed vs centralized database, role of dba.

## UNIT-IV

**Data classification:** importance of data, private organizations versus military classifications: threats and risk: confidentially, authentication, integrity, non-repudiation, cryptography: type of cryptogram, symmetric key cryptography, and asymmetric key cryptography: digital signature.

## References:

1. An introduction to database systems by Bipin Desai, Galgotia Pub.

2. An introduction to database management systems by C.J. Date, Pearson Education,7th edition.
3. Database System Concepts by Silberschatz Abraham Korth, TMH,4th edition.

---

# 17MSE23C3-LITERATURE SURVEY-(DISSERTATION STAGE 1)

| | |
|---|---|
| L, T, P: 0, 0, 2 | Class Work: 100 Marks |
| | Total: 100 Marks |

**COURSE OUTCOMES:**

By the end of this course every student is expected to be able to

CO1 understand the process of research.
CO2 do literature survey to identify a research problem.
CO3 communicate and discuss research ideas.
CO4 plan and write dissertation synopsis.

-------------------------------------------------------------------------------------------------------------

# 17MSE23C4-SEMINAR

L, T, P: 0, 0, 2 Class work: 50 Marks
Total: 50 Marks

At the end of this course the student shall be able to
CO1 prepare the topic and contents on a technical topic
CO2 speak on a technical topic effectively
CO3 enhance communication skills

## 17MSE23CL1-ADVANCE DATABASE MANAGEMENT LAB

| | |
|---|---|
| L, T, P: 0, 0, 2 | Sessional: 50 Marks |
| | Exam: 50 Marks |
| | Total: 100 Marks |

Practical's based on theory paper

### Course Outcomes

By the end of this course the student will have

CO1 Ability to select the data structures that efficiently model the information in a problem.

CO2 Understand the fundamentals of Relational database design and normal forms

CO3 Master the basics of my SQL for retrieval and management of data.

CO4 Ability to understand the basics of transaction processing and concurrency control.

-----------------------------------------------------------------------------------------------------------

## 17MSE23CL2-Project

L, T, P: 0, 0, 1

Sessional: 50 Marks
Exam: 50 Marks
Total: 100 Marks

## Course Outcomes

At the end of the course, student would be able to

CO1 Synthesize and apply prior knowledge to designing and implementing solutions to open- ended computational problems while considering multiple realistic constraints.

CO2 Design and develop the software with SE practices and standards

CO3 Analyze database, network and application design methods

CO4 Analyzing professional issues, including ethical, legal and security issues, related to computing projects.

**17MSE24C1 DISSERTATION-II (IV sem)**

COURSE OUTCOMES:

By the end of this course every student is expected to be able to

CO1 handle research problems and use modern research tools/methods.
CO2 analyse and review the existing literature on a research problem.
CO3 design and conduct experiments.
CO4 write dissertation and technical reports.
CO5 publish research papers.