ANDHRA UNIVERSITY

DEPARTMENT OF INFORMATION TECHNOLOGY

PROGRAM : M.SC(COMPUTER SCIENCE)
REGULATION AND SYLLABUS
EFFECTIVE FROM 2016-2017 BATCH
# Master of Science in Computer Science (M.Sc.)

Course Structure and Scheme of Valuation w.e.f. 2016-17

## I SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>Name of the subject</th>
<th>Periods/week</th>
<th>Max. Marks</th>
<th>Total</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
<td>Lab</td>
<td>Ext.</td>
<td>Int.</td>
</tr>
<tr>
<td>MSCS 1.1</td>
<td>Discrete Mathematical Structures</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.2</td>
<td>Data Structures &amp; File Structures</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.3</td>
<td>Computer Organization &amp; Architecture</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.4</td>
<td>Object oriented Programming using C++ &amp; JAVA</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.5</td>
<td>Advanced Computer Networks</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.6</td>
<td>Data &amp; File Structures Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>MSCS 1.7</td>
<td>Computer Organization Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>20</td>
<td>6</td>
<td>450</td>
<td>250</td>
</tr>
</tbody>
</table>
## II SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>Name of the subject</th>
<th>Periods/week</th>
<th>Max. Marks</th>
<th>Total</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
<td>Lab</td>
<td>Ext.</td>
<td>Int.</td>
</tr>
<tr>
<td>MSCS 2.1</td>
<td>Formal Languages &amp; Automata Theory</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.2</td>
<td>Relational Data Base Management Systems</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.3</td>
<td>Advanced Operating Systems</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.4</td>
<td>Elective I</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.5</td>
<td>Elective-II</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.6</td>
<td>Advanced JAVA Programming Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>MSCS 2.7</td>
<td>Relational Data Base Management Systems Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>20</td>
<td>6</td>
<td>450</td>
<td>250</td>
</tr>
</tbody>
</table>
Master of Science in Computer Science (M.Sc.)
Course Structure and Scheme of Valuation w.e.f. 2016-17

III SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>Name of the subject</th>
<th>Periods/week</th>
<th>Max. Marks</th>
<th>Total Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSCS 3.1</td>
<td>Data Warehousing &amp; Mining</td>
<td>4</td>
<td>70 30</td>
<td>100</td>
</tr>
<tr>
<td>MSCS 3.2</td>
<td>Object Oriented Software Engineering</td>
<td>4</td>
<td>70 30</td>
<td>100</td>
</tr>
<tr>
<td>MSCS 3.3</td>
<td>Network Security &amp; Cryptography</td>
<td>4</td>
<td>70 30</td>
<td>100</td>
</tr>
<tr>
<td>MSCS 3.4</td>
<td>Elective III</td>
<td>4</td>
<td>70 30</td>
<td>100</td>
</tr>
<tr>
<td>MSCS 3.5</td>
<td>MOOCS-I</td>
<td></td>
<td>--</td>
<td>100</td>
</tr>
<tr>
<td>MSCS 3.6</td>
<td>OOSE Lab</td>
<td>--</td>
<td>3</td>
<td>50 50</td>
</tr>
<tr>
<td>MSCS 3.7</td>
<td>Network Programming &amp; Web Programming Lab</td>
<td>--</td>
<td>3</td>
<td>50 50</td>
</tr>
<tr>
<td>MSCS 3.8</td>
<td>Seminar on Advanced Topics</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td>16</td>
<td>6</td>
<td>450</td>
</tr>
</tbody>
</table>

Elective III: Artificial Intelligence / Operations Research / Cloud computing/ BigData Analytics

MOOCS-I:
Each student should learn any one of the following topics by registering for courses through Online instruction from standard e-learning portals like nptel, coursera, etc. and write the examination conducted as per the university norms.

List of topics for MOOCS-I:
Internet of Things/ R Programming for Data Analytics/ Data Visualization using Tableau/ MongoDB for Developers/ DevOps/Agile Technologies for Software Development
Master of Science in Computer Science (M.Sc.)
Course Structure and Scheme of Valuation w.e.f. 2016-17

IV SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>NAME OF THE SUBJECT</th>
<th>MAXIMUM MARKS</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>INTERNAL</td>
<td>EXTERNAL</td>
</tr>
<tr>
<td>MSCS 4.1</td>
<td>PROJECT/ THESIS WORK</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Total Credits (Complete Course) 87

Instructions for Project

1. Three Stages In Project adjudication:

   Stage I: Presentation of Concept Note & Problem Approval by Guide
   Stage II: Progress Approval by System Demonstration with results Internal -50 Marks
   Stage III: Final Presentation with Documentation & External Viva-Voce - 50 Marks

2. Candidates can do their thesis work within the department or in any industry/research organization for two semesters (4th semesters). In case of thesis done in an industry/research organization, one advisor (Guide) should be from the department and one advisor (CO-Guide) should be from the industry/research organization.

3. A publication of a paper on the thesis work in a National/International Conference proceedings with presentation certificate or a paper on the thesis work be communicated to a National/International Journal & accepted for publication for the submission of thesis at the end of 4th semester is desirable.

4. The external examiner shall be nominated by the Chairman, Board of Examiners in CSSE as per the norms of the University.
M.Sc – COMPUTER SCIENCE

Program Educational Objectives (PEOs)

PEO 1: To Transcend in professional career and/or pursue higher education and research utilizing the knowledge gained in computational domain, mathematics, and management.
PEO 2: To analyze real world problems, develop feasible and environmentally acceptable solutions to achieve peer recognition as an individual or in a team.
PEO 3: To Work in multidisciplinary environment with ethical and sustainable computing perspectives, adaptable to the changing trends in technology and society by engaging in lifelong learning.
PEO 4: To Identify opportunity to evolve as an entrepreneur and pursue the same for the benefit of individual and society.

Program Specific Outcomes (PSOs)

PSO 1: Analyze, Design, Test and Implement components, processes and solutions for specific application development using appropriate data modeling concepts and documentation.
PSO 2: Adapt and use appropriate modern software tools, resources and techniques to solve real-world problems within the framework of constraints.
PSO 3: Apply concepts of core computer science to build and manage solutions for enterprises.

Program Outcomes (PO)

PO 1: Apply knowledge of computing fundamentals, mathematics, and domain knowledge for solving different problems.
PO 2: Design and evaluate solutions for complex computing problems, and design and evaluate systems, components, or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.
PO 3: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO 4: Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations.

PO 5: Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice.

PO 6: Recognize the need, and have the ability, to engage in independent learning for continual development as a computing professional.

PO 7: Demonstrate knowledge and understanding of the computing and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 8: Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions.

PO 9: Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice.

PO 10: Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments.

PO 11: Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.
Master of Science in Computer Science (M.Sc.)

MSCS 1.1  Discrete Mathematical Structures

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits:4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
</tbody>
</table>

Course Objectives:
1. Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic.
2. Solve problems using Set Laws, Operations and properties of relations, Functions and Types of functions and Recursive Functions.
3. Solve counting problems by applying elementary counting techniques using the product and sum rules, permutations, combinations, the pigeon-hole principle, and binomial expansion.
4. Definition and identify different types of Graphs, Trees and Minimal spanning trees and Tree traversal Algorithms.

Course Outcomes:
1. Ability to apply the rules and laws of propositional logic on statements.
2. Understands the basic principles and operations on sets.
3. Attains capability to solve recursive functions and permutations and combinations.
4. Ability to understand graph theory and its applications.
5. Obtains knowledge in applications of trees.

MSCS 1.2  Data Structures & File Structures

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits:4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
</tbody>
</table>

Course Objectives:
1. Exploring basic data structures such as stacks and queues.
2. Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs.
3. Introduces sorting and pattern matching algorithms

Course Outcomes:
1. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.
3. Implement and know the application of algorithms for sorting and pattern matching.
4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees.
MSCS 1.3  
**Computer Organization & Architecture**  
Instruction: 3 Periods + 1 Tut/week  
Credits: 4  
Internal: 30 Marks  
University Exam: 70 Marks  
Total: 100 Marks

**Course Objectives:**
1. To study about structure and functional components of a computer.
2. Understanding the hierarchical organization of a computer system which consists of instruction set of commands.
3. Learn about the architecture of a computer from a programming view.
4. To design a balance system that minimizes performance and utilization of all elements.

**Course Outcomes:**
1. Understands about data representation and computer arithmetic.
2. Acquires knowledge on Boolean Algebra and 8085 instruction set architecture.
3. Understands the basics of computer organization.
4. Ability to understand and design CPU of a computer.
5. Ability to analyze the input and output organization of a computer.

---

MSCS 1.4  
**Object oriented Programming Using C++ & JAVA**  
Instruction: 3 Periods + 1 Tut/week  
Credits: 4  
Internal: 30 Marks  
University Exam: 70 Marks  
Total: 100 Marks

**Course Objectives:**
1. Introduces Object Oriented Programming concepts using the C++ language.
2. Introduces the principles of data abstraction, inheritance and polymorphism;
3. Introduces the principles of virtual functions and polymorphism
4. Introduces handling formatted I/O and unformatted I/O
5. Introduces exception handling
6. To introduce the principles of inheritance and polymorphism; and demonstrate how they relate to the design of abstract classes
7. To introduce the implementation of packages and interfaces
8. To introduce the concepts of exception handling and multithreading.

**Course Outcomes:**
1. Able to develop programs with reusability
2. Develop programs for file handling
3. Handle exceptions in programming
4. Develop applications for a range of problems using object-oriented programming techniques
5. Able to solve real world problems using OOP techniques.
6. Able to understand the use of abstract classes.
7. Able to solve problems using java collection framework and I/o classes.
MSCS 1.5  
**Advanced Computer Networks**

**Instruction:** 3 Periods + 1 Tut/week  
**Credits:** 4

| Internal: 30 Marks | University Exam: 70 Marks | **Total:** 100 Marks |

**Course Objective:**
1. To prepare students to know the characteristics and designs of types of computer networks and their applications.
2. Learn how computer network hardware and software operate.
3. Investigate the fundamental issues driving network design.
4. Learn about dominant network technologies.

**Course Outcomes:**
1. Understands the overview of Data Communications and Networks.
2. Performs a thorough study of physical and data link layers.
3. Familiarizes with frame formats of data link layer.
4. Gains knowledge about network and transport layer functionalities.
5. Learns practical applications of networks.

MSCS 1.6  
**Advanced Computer Networks Lab**

**Instruction:** 3 Periods/week  
**Credits:** 2

| Internal: 50 Marks | University Exam: 50 Marks | **Total:** 100 Marks |

**Course Objectives:**
1. To implement Applications using C++.
2. To develop programs for searching and sorting algorithms.
3. To write programs using concepts of various trees.
4. To implement programs using graphs.

**Course Outcomes:**
1. Gains basic programming skills in C++.
2. Ability able to write programs to implement stacks and queues.
4. Ability to implement programs using trees and graphs.
5. Develops skills in designing applications using data structures.

MSCS 1.7  
**Computer Organization Lab**

**Instruction:** 3 Periods/week  
**Credits:** 2

| Internal: 50 Marks | University Exam: 50 Marks | **Total:** 100 Marks |

**Course Objectives:**
1. To learn the about logic gates, half adders, full adders and flip-flops.
2. To learn about the microprocessor programming.
3. To learn about the microprocessor interfacing with stepper motor, R-2R ladder.
4. To develop the skill in writing microprocessor programming.

**Course Outcomes:**
1. The student understands and learns the applications of Digital logic design.
2. The student understands and learns the concept of memory design.
3. The student understands and learns the concept of data interpretation.
4. The student understands and learns the concept of data transmission.
5. The student develops the skill of writing microprocessor programming.
MSCS 2.1  FORMAL LANGUAGES & AUTOMATA THEORY

Instruction: 3 Periods + 1 Tut/week

| Internal: 30 Marks | University Exam: 70 Marks | Total: 100 Marks |

Course Objectives:
1. Understand basic properties of Deterministic and Nondeterministic Finite Automata.
2. Understand basic properties of Languages, Grammars, Normal forms and difference between types of languages and types of automata.
4. Understand the challenges of Theoretical computer science and its contribution to other sciences.

Course Outcomes:
1. Familiarizes with various types of Finite Automata.
2. Understand the types of Grammar and Regular expressions.
3. Learn the concepts of Context Free Language, Normal Forms and Pushdown Automata.
4. Ability to construct Turing machines and apply on its applications.

MSCS 2.2  RELATIONAL DATA BASE MANAGEMENT SYSTEMS

Instruction: 3 Periods + 1 Tut/week

| Internal: 30 Marks | University Exam: 70 Marks | Total: 100 Marks |

Course Objectives:
1. To understand the different issues involved in the design and implementation of a database system.
2. To study the physical and logical database designs, database modeling, relational, hierarchical, and network models.
3. To understand and use data manipulation language to query, update, and manage a database.
4. To develop an understanding of essential DBMS concepts such as: database security, integrity, and concurrency.
5. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.

Course Outcomes:
1. Understands various database models.
2. Obtain querying techniques in Entity Relation model.
3. Learn optimization of database design with Normalization.
4. Familiarizes with the concepts of Serializability, Concurrency control and crash recovery.
5. Gain an overview of storage and indexing structures

MSCS 2.3  ADVANCED OPERATING SYSTEMS

Instruction: 3 Periods + 1 Tut/week

| Internal: 30 Marks | University Exam: 70 Marks | Total: 100 Marks |

Course Objectives:
1. To learn about fundamentals, functions and types of operating systems.
2. To learn the mechanisms of OS to handle processes and threads and their communication.
3. To gain knowledge on operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and memory management algorithms.
4. To provide the knowledge towards process synchronization and virtual memory, Secondary storage structure and Security.
Course Outcomes:
1. Familiarizes with the fundamentals and different types of operating systems.
2. Ability to learn Process Scheduling and synchronization.
3. Acquaint knowledge about Deadlocks.
4. Learns about memory management and CPU scheduling techniques.
5. Studies about Disk Scheduling, Disk Management and Security issues

<table>
<thead>
<tr>
<th>MSCS 2.4</th>
<th>Elective I</th>
<th>DESIGN AND ANALYSIS OF ALGORITHMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction: 3 Periods + 1 Tut/week</td>
<td>Credits:4</td>
<td></td>
</tr>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
<td>Total: 100 Marks</td>
</tr>
</tbody>
</table>

Course Objectives:
1. Introduces the notations for analysis of the performance of algorithms.
2. Introduces the data structure disjoint sets.
3. Describes major algorithmic techniques (divide-and-conquer, backtracking, dynamic programming, greedy, branch and bound methods) and mention problems for which each technique is appropriate;
4. Describes how to evaluate and compare different algorithms using worst-, average-, and bestcase analysis.
5. Explains the difference between tractable and intractable problems, and introduces the problems that are P, NP and NP complete.

Course Outcomes:
1. Ability to analyze the performance of algorithms
2. Ability to choose appropriate data structures and algorithm design methods for a specified application
3. Ability to understand how the choice of data structures and the algorithm design methods impact the performance of programs.

<table>
<thead>
<tr>
<th>MSCS 2.4</th>
<th>Elective I</th>
<th>IMAGE PROCESSING</th>
</tr>
</thead>
<tbody>
<tr>
<td>Instruction: 3 Periods + 1 Tut/week</td>
<td>Credits:4</td>
<td></td>
</tr>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
<td>Total: 100 Marks</td>
</tr>
</tbody>
</table>

Course Objectives
1. Provide a theoretical and mathematical foundation of fundamental Digital Image Processing concepts.
2. The topics include image acquisition; sampling and quantization; preprocessing; enhancement; restoration; segmentation; and compression.

Course Outcomes
1. Demonstrate the knowledge of the basic concepts of two-dimensional signal acquisition, sampling, and quantization.
2. Demonstrate the knowledge of filtering techniques.
3. Demonstrate the knowledge of 2D transformation techniques.
4. Demonstrate the knowledge of image enhancement, segmentation, restoration and compression techniques.
MSCS 2.4  Elective I  EMBEDDED SYSTEMS

Instruction: 3 Periods + 1 Tut/week  Credits:4
Internal: 30 Marks  University Exam: 70 Marks  Total: 100 Marks

Course Objectives:
1. To study the basics of embedded systems and its examples.
2. To study the 8051 Microcontroller architecture and its instruction set.
3. To discuss various software architectures in embedded systems.
4. To discuss Inter Task Communication procedures in RTOS and design issues of RTOS.
5. To study various embedded software development tools and debugging techniques.

Course Outcomes:
1. Understands the basics of Embedded systems, Microprocessors and Microcontrollers.
2. Develops ability to write programs using 8051 Assembly Language instructions.
3. Learns about various Interrupts and Software Architecture.
4. Analyzes various design issues of RTOS.
5. Familiarizes with embedded software development tools and debugging techniques.

MSCS 2.5  Elective II  WEB TECHNOLOGIES

Instruction: 3 Periods + 1 Tut/week  Credits:4
Internal: 30 Marks  University Exam: 70 Marks  Total: 100 Marks

Course Objectives:
1. Understand the principles of Web based application development.
2. Design dynamic content in Web Pages using JavaScript and XML.
3. Understanding the concepts of java Servlets, java Server Pages and design applications using them.
4. Understand the concepts of Component development and design applications by establishing connections to Databases.
5. Understand the concepts of PHP and design applications.

Course Outcomes:
1. Ability to construct web-based applications using Java script and XML.
2. Learns to design application using java Servlets.
3. Develops competency to design sophisticated Java Server Pages.
4. Understands the concepts of JDBC connectivity.
5. Gains knowledge on designing applications using PHP.

SCS 2.5  Elective II  Mobile Computing

Instruction: 3 Periods + 1 Tut/week  Credits:4
Internal: 30 Marks  University Exam: 70 Marks  Total: 100 Marks

Course Objectives:
2. To learn about the concepts and principles of mobile computing;
3. To explore both theoretical issues of mobile computing;
4. To develop skills of finding solutions and building software for mobile computing applications.
Course Outcomes
1. Acquires concepts and features of cellular technologies and mobile services.
2. Gains knowledge on Wireless-LAN’s and their standards.
3. Identifies the important issues of wireless networks and protocol mechanisms.
4. Learns the functionalities of database in mobile communications and issues.

MSCS 2.5  Elective II  Wireless Sensor Networks

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
<tr>
<td>Total: 100 Marks</td>
<td></td>
</tr>
</tbody>
</table>

Course Objectives:
1. To understand the concepts of sensor networks
2. To understand the MAC and transport protocols for ad hoc networks
3. To understand the security of sensor networks
4. To understand the applications of adhoc and sensor networks

Course Outcomes:
1. Ability to understand the state-of-the-art research in the emerging subject of Ad Hoc and Wireless Sensor Networks
2. Ability to solve the issues in real-time application development based on ASN.
3. Ability to conduct further research in the domain of ASN
4. Acquaint with the knowledge on routing protocols.
5. Acquires knowledge on issues and challenges on Quality of Services.
6. Understands the architecture of wireless sensor networks.

MSCS 2.6  Advanced JAVA Programming Lab

<table>
<thead>
<tr>
<th>Instruction: 3 Periods/week</th>
<th>Credits: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 50 Marks</td>
<td>University Exam: 50 Marks</td>
</tr>
<tr>
<td>Total: 100 Marks</td>
<td></td>
</tr>
</tbody>
</table>

Course Objectives:
1. To write programs using abstract classes.
2. To write programs for solving real world problems using java collection frame work.
3. To write multithreaded programs.
4. To write GUI programs using swing controls in Java.
5. To introduce java compiler and eclipse platform.
6. To impart hands on experience with java programming

Course Outcomes:
1. Able to write programs for solving real world problems using java collection frame work.
2. Able to write programs using abstract classes.
3. Able to write multithreaded programs.
4. Able to write GUI programs using swing controls in Java.
Relational Data Base Management Systems Lab

<table>
<thead>
<tr>
<th>Instruction: 3 Periods/week</th>
<th>Credits: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 50 Marks</td>
<td>University Exam: 50 Marks</td>
</tr>
</tbody>
</table>

**Course Objectives:**
1. The major objective of this lab is to provide a strong formal foundation in database concepts, technology and practice to the participants to groom them into well informed database application developers.
2. To present SQL and procedural interfaces to SQL comprehensively
3. To present the concepts and techniques relating to query processing by SQL Engines.
4. To understand and use data manipulation language to query, update, and manage a Database.
5. To present the concepts and techniques relating to ODBC and its Implementations.
6. To design and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS

**Course Outcomes:**
1. Practices DDL, DML, DCL commands.
2. Design and implement a database schema for a given problem-domain and normalize a database.
3. Declare and enforce integrity constraints on a database using a state-of-the-art RDBMS.
4. Practice PL/SQL programming.
5. Familiarizes with database connectivity.

---

Data Warehousing & Data Mining

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
</tbody>
</table>

**Course Objectives:**
1. To understand the evolution of data warehousing and data mining systems
2. To understand extracting, cleaning and transformation of data into a warehouse.
3. To learn the principles of statistics, information theory, machine learning, AI and implementation of data mining techniques.
4. To understand pattern mining using classification and clustering methods

**Course Outcomes:**
1. Learns about data mining concepts and functionalities.
2. Familiarizes with various data preprocessing techniques.
3. Gains knowledge about association rule mining techniques.
4. Understands Classification and Prediction techniques.
5. Analyzes Clustering techniques.
### MSCS 3.2

**Object Oriented Software Engineering**

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
</tbody>
</table>

**Course Objectives**

1. The aim of the course is to provide an understanding of the working knowledge of the techniques for estimation, design, testing and quality management of large software development projects.
2. Topics include process models, software requirements, software design, software testing, software process/product metrics, risk management, quality management and UML diagrams.

**Course Outcomes**

1. Ability to translate end-user requirements into system and software requirements, using e.g. UML, and structure the requirements in a Software Requirements Document (SRD).
2. Identify and apply appropriate software architectures and patterns to carry out high level design of a system and be able to critically compare alternative choices.
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report.

### MSCS 3.3

**NETWORK SECURITY & CRYPTOGRAPHY**

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
</tbody>
</table>

**Course Objectives:**

1. Introduction of the importance of various services of network security.
2. To discuss various cryptographic techniques.
3. Exploration of different types of security threats and remedies and understanding of internet security protocols and standards.
4. To introduce types of malicious software and issues.

**Course Outcomes:**

1. Learns and understands the importance of cryptography.
2. Familiarizes with the algorithms of various security services.
3. Ability to understand various key management and authentication techniques.
4. Understands various cryptographic algorithms for e-mail security and transport-level security.
5. Gains knowledge about IP-security, malicious software and related attacks.

### MSCS 3.4

**Elective-III  ARTIFICIAL INTELLIGENCE**

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
</tbody>
</table>

**Course Objectives:**

1. The primary objective of this course is to introduce the basic principles, techniques, and applications of Artificial Intelligence that includes problem solving, Searching Techniques, knowledge representation, logics, reasoning, planning, perception & action, and learning.
2. To learn about AI problem, Production Systems and their characteristics.
3. To understand the importance of search and the corresponding search strategies for solving AI problem.
4. Become familiar with basic principles of AI toward knowledge representation, logic and Reasoning.
5. Investigate applications of AI techniques in intelligent agents, expert systems, artificial neural networks.
Course Outcomes:
1. Understands the history of Artificial Intelligence and its foundations.
2. Familiarizes with knowledge representation issues and concepts.
3. Obtains the knowledge to represent the language sentences using predicate logic.
4. Gains awareness about expert system.
5. Develops awareness on neural networks models.

MSCS 3.4  Elective-III  OPERATIONS RESEARCH

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
<tr>
<td></td>
<td>Total: 100 Marks</td>
</tr>
</tbody>
</table>

Course Objectives
1. Formulate a real-world problem as a mathematical programming model
2. Implement and solve the model in EXCEL and LINDO
3. Understand the theoretical workings of the simplex method for linear programming and perform iterations of it by hand
4. Understand the relationship between a linear program and its dual, including strong duality and complementary slackness
5. Perform sensitivity analysis to determine the direction and magnitude of change of a model’s optimal solution as the data change
6. Solve specialized linear programming problems like the transportation and assignment problems
7. Solve network models like the shortest path, minimum spanning tree, and maximum flow problems

Course Outcomes
By the end of the course, the student will be able to:
1. Students will be able to describe characteristics and scope of OR.
2. Students will be able to define and formulate mathematical problems.
3. Students will be able to select optimal problems solving techniques for a given problem using LP.
4. Students will be able to formulate and solve transportation, travelling salesman and transshipment problems.
5. Students will be able to formulate and solve optimization problems related to job/ work assignments.
6. Students will be able to demonstrate and solve simple models of Game theory.
7. Students will be able to evaluate optimum solution using dynamic programming for different applications.
8. Students will be able to choose / devise appropriate queuing model for practical application.
9. Students will be able to solve different problems related to Network
MSCS 3.4 | Elective-III | CLOUD COMPUTING
--- | --- | ---
**Instruction:** 3 Periods + 1 Tut/week | **Credits:** 4
**Internal:** 30 Marks | **University Exam:** 70 Marks | **Total:** 100 Marks

**Course Objectives:**
1. The objective of this course is to gain the in-depth knowledge of Cloud Computing concepts, technologies, architecture and applications.
2. To introducing and researching state-of-the-art in Cloud Computing fundamental issues, technologies, applications and implementations.
3. Another objective is to expose the students to frontier areas of Cloud Computing.

**Course Outcomes:**
1. Understands cloud computing platforms and their migration issues.
2. Learns about Virtual Machines Provisioning and Scheduling Techniques.
3. Gains knowledge on Integration of Private and Public Clouds.
5. Develops the knowledge of Architecting Cloud Applications in the AWS and Cloud Mashups.

MSCS 3.4 | Elective-III | BIGDATA ANALYTICS
--- | --- | ---
**Instruction:** 3 Periods + 1 Tut/week | **Credits:** 4
**Internal:** 30 Marks | **University Exam:** 70 Marks | **Total:** 100 Marks

**Course Objectives:**
On completing this course student will be able to
1. Understand big data and Apache Hadoop Eco system
2. Understand distributed, parallel, cloud computing and SQL concepts
3. Apply Hadoop concepts
4. Understand concepts of map and reduce and functional programming

**Course Outcomes:**
1. Gain conceptual understanding of analytics concepts, algorithms and statistical tests
2. Students will be able to look at the core projects used for both batch and real time data processing such as Hadoop
3. Students will be able to look at wider range of problems and data science based solutions

MSCS 3.6 | Object Oriented Software Engineering Lab
--- | ---
**Instruction:** 3 Periods/week | **Credits:** 2
**Internal:** 50 Marks | **University Exam:** 50 Marks | **Total:** 100 Marks

**Course Objectives:**
1. To have hands on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

**Course Outcomes:**
1. Ability to translate end-user requirements into system and software requirements
2. Ability to generate a high-level design of the system from the software requirements
3. Will have experience and/or awareness of testing problems and will be able to develop a simple testing report
MSCS 3.7  
**Network Programming Lab**

<table>
<thead>
<tr>
<th>Instruction: 3 Periods/week</th>
<th>Credits: 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 50 Marks</td>
<td>University Exam: 50 Marks</td>
</tr>
</tbody>
</table>

**Course Objectives:**
1. To understand inter process and inter-system communication
2. To understand socket programming in its entirety
3. To understand usage of TCP/UDP / Raw sockets
4. To understand how to build network applications

**Course Outcomes:**
1. To write socket API based programs
2. To design and implement client-server applications using TCP and UDP sockets
3. To analyze network programs

MSCS 3.8  
**Seminar on Advanced Topics**

<table>
<thead>
<tr>
<th>Instruction: 3 Periods/week</th>
<th>Credits: 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 50 Marks</td>
<td>University Exam: 50 Marks</td>
</tr>
</tbody>
</table>

**Course Objectives:**
To enable a student to be familiar with Communication skill

**Course Outcomes:**
Student is expected to learn how to make a presentation, how to write a report and group discussions.
Master of Science in Computer Science (M.Sc.)  
Course Structure and Scheme of Valuation w.e.f. 2016-17

I SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>Name of the subject</th>
<th>Periods/week</th>
<th>Max. Marks</th>
<th>Total</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
<td>Lab</td>
<td>Ext.</td>
<td>Int.</td>
</tr>
<tr>
<td>MSCS 1.1</td>
<td>Discrete Mathematical Structures</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.2</td>
<td>Data Structures &amp; File Structures</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.3</td>
<td>Computer Organization&amp; Architecture</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.4</td>
<td>Object oriented Programming using C++ &amp; JAVA</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.5</td>
<td>Advanced Computer Networks</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 1.6</td>
<td>Data &amp; File Structures Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>MSCS 1.7</td>
<td>Computer Organization Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>20</td>
<td>6</td>
<td>450</td>
<td>250</td>
</tr>
</tbody>
</table>
1. **Sets, relations and functions:** Operations on sets, relations and functions, binary relations, partial ordering relations, equivalence relations, principles of mathematical induction.

2. Permutations and combinations; recurrence relation and generating functions.

3. **Algebraic structures and morphisms:** Algebraic structures with one binary operation - semigroups, monoids and groups, congruence relation and quotient structures. Free and cyclic monoids and groups, permutation groups, substructures, normal subgroups.


5. **Mathematical logic:** Syntax, semantics of Propositional and predicate calculus, valid, satisfiable and unsatisfiable formulas, encoding and examining the validity of some logical arguments.

6. **Proof techniques:** forward proof, proof by contradiction, contrapositive proofs, proof of necessity and sufficiency.

7. **Graph Theory:** Graphs and digraphs, trees, Eulerian cycle and Hamiltonian cycle, adjacency and incidence matrices, vertex colouring, planarity.

**Text Book**


**Reference Books:**

MSCS 1.2

Data Structures & File Structures

Instruction: 3 Periods + 1 Tut/week
Credits: 4

Internal: 30 Marks
University Exam: 70 Marks
Total: 100 Marks

PART - A

Introduction to Data Structures

The Stack: Primitive operations – As an Abstract Data Type – Implementing the Stack operations using Arrays, and Structures

Queues:: The Queue as Abstract Data Type – Sequential Representation ,Types of Queues – Operations – Implementation using Arrays, and Structures


Tree Searching: Insertion into a Binary Search Tree – Deleting from a Binary Search Tree – Efficiency of Binary Search Tree operation

PART - B

File Processing Operations

Physical and logical files, opening, reading & writing and closing files in C, seeking and special characters in files, physical devices and logical files, file-related header files in C

Secondary Storage

Disks – organization, tracks, sectors, blocks, capacity, non-data overhead, cost of a disk access,Magnetic Tape – types, performance, organization estimation of tape length and data transmission times

Journey and buffer Management

File manager, I/O buffer, I/O processing, buffer strategies and bottlenecks

File Structure Concepts

A stream file, field structures, reading a stream of fields, record structures and that uses a length indicator, Mixing numbers and characters – use of a hex dump, reading the variable length records from the files

Managing records in C files

Retrieving records by keys, sequential search, direct access, choosing a record structure and record length, header records, file access and file organization

Organizing files for performance
Data compression, reclaiming space – record deletion and storage compaction, deleting fixed-length records for reclaiming space dynamically, deleting variable-length records, space fragmentation, replacement strategies.

**Indexing**

Index, A simple index with an entry sequenced file, basic operations on an indexed, entry sequenced file, indexes that are too large to hold in memory, indexing to provide access by multiple keys, retrieval using combination of secondary keys, improving the secondary index structure – inverted lists

**Indexed sequential file access and prefix B+ Trees**

Indexed sequential access, maintaining a sequence set, adding a simple index to the sequence set, the content of the index: separators instead of keys, the simple prefix B+ tree, simple prefix B+ tree maintenance, index set block size, internal set block size, internal structure of index set blocks: a variable order B-tree, loading a simple prefix B+ tree

**Hashing**

Collisions in hashing, a simple hashing algorithms, hashing functions and record distributions, memory requirements, collision resolution by progressive overflow, buckets, deletions.

**Textbooks:**

3. File Structures – An Object Oriented Approach with C++ by Michael J. Folk, Bill Zoellick and Greg Riccardi, Pearson
1. **Register Transfer and Micro operations:**
   Register Transfer Language, Register Transfer, Bus and Memory Transfers, Arithmetic Micro operations, Logic Micro operations, Shift Micro operations, Arithmetic Logic Shift Unit.

2. **Basic Computer Organization and Design:**
   Instruction Codes, Computer Registers, Computer Instructions, Timing and Control, Instruction Cycle, Memory-Reference Instructions, Input-Output and Interrupt, Complete Computer Description, Design of Basic Computer, Design of Accumulator Logic.

3. **Micro programmed Control:**
   Control Memory, Address Sequencing, Micro program Example, Design of Control Unit.

4. **Central Processing Unit:**
   Introduction, General Register Organization, Stack Organization, Instruction Formats, Addressing Modes, Data Transfer and Manipulation, Program Control, Reduced Instruction Set Computer (RISC)

5. **Input/output Organization:**
   Peripheral Devices, I/O interface, Asynchronous data transfer, Modes of transfer, priority Interrupt, Direct memory access, Input-Output Processor (IOP), Serial Communication.

6. **Memory Organization:**
   Memory Hierarchy, Main memory, Auxiliary memory, Associate Memory, Cache Memory, and Virtual memory, Memory Management Hardware.

7. **Overview of Computer Architecture:**

**Text Book:**


Reference Book:


MSCS 1.4  

Object oriented Programming Using C++ & JAVA

Instruction: 3 Periods + 1 Tut/week  
Credits:4

Internal: 30 Marks  University Exam: 70 Marks  Total: 100 Marks


2. **Basic OF C ++**: Structure Of C++ Program, Data Types And Declaration, Expressions And Operator Precedence, Program Flow Control, Functions, Scope of Variables, Inline Functions and Default Arguments, Dynamic Allocation New And Delete Operators.

3. **Classes & Objects**: Classes And Objects, User Defined Data Types, Constructors & Destructors, Controlling and Accessibility, Class Members, Member Functions, Friend Functions, This Pointer, Static and Constant Member Functions, Type Conversions, Function selection, Friend Functions, Function Adapters, String Library, Class Templates, Function Templates, Member Function Templates.

4. **Inheritance & Polymorphism**: Derived Classes, Syntax Of Derived Classes, Making Private Members Inheritable, Single, Multilevel, Multiple, Hierarchical, Hybrid Inheritance, Polymorphism, Operator Overloading, Function Overloading, Pointers, pointers to objects, this pointer, pointers to derived classes, virtual and pure virtual functions.

5. **Introduction to JAVA**: Applets & Applications, Structure of Java Program, Introduction to Classes and Objects, Arrays, strings and Vectors.

6. **Packages and Interface, and Multi threading**: Packages, Interfaces, creating, extending, stopping, blocking threads, thread states, thread methods, exceptions, priority in threads, synchronization, Runnable interface.

7. **Exception Handling In C++ & Java**: Exception, Handling, Throwing Exceptions, Try Blocks, Handlers, Exception Specification, Standard Exceptions And Uses.


**Text Books:**

1. Object Oriented Programming in C++, Robert Lafore
2. Introduction to JAVA PROGRAMMING by Y.Daniel Liang (PHI)

**References:**

1. Object Oriented Programming in C++: N. Barkakati, PHI
2. Object Oriented Programming using C++, Ira Pohl, PEARSON EDUCATIO
3. JAVA 2.0- Complete Reference : Herbert Schildt & F. Naughton.
4. Object oriented Programming using C++: E. Balagurusamy, PHI.
5. Programming with JAVA- A primer: E. Balagurusamy, PHI
1. **Introduction to Computer Networks:** Introduction, Network Hardware, Network Software, Reference Models, Data Communication Services & Network Examples, Internet Based Applications.

2. **Data Communications:** Transmission Media, Wireless Transmission, Multiplexing, Switching, Transmission in ISDN, Broad Band ISDN, ATM Networks,


5. **Internet Transport Protocols:** TRANSPORT Service, Elements of Transport Protocols, TCP and UDP Protocols, Quality of Service Model, Best Effort Model, Network Performance Issues.


7. **Network Devices:** Over View of Repeaters, Bridges, Routers, Gateways, Multi protocol Routers, Brouters, Hubs, Switches, Modems, Channel Service Unit CSU, Data Service Units DSU, NIC, Wireless Access Points, Transceivers, Firewalls, Proxies.


**Text Book:**


**References:**

2. Computer networks, Mayank Dave, CENGAGE.
Implementation of following problems using C++

1. To implement various operations such as insertion, deletion, display on the following ADTs
   a) Stack           b) Queues       c) Single linked lists.

2. To perform the following using ADT Stack
   a) To convert the given infix expression to postfix expression
   b) To evaluate a postfix expression

3. To implement insert, delete, display, traverse forward and traverse backward on a doubly linked list

4. To implement insert, delete, inorder, postorder and preorder traversals on a binary search tree.

5. Implement the following simple file manipulations
   a) to read a text file and copy it character by character to another output file.
   b) to check whether a given file exists or not.
   c) to read numeric data from a text file.
   d) to create a text file, write sentences into it from the keyboard.
   e) to copy a source file, line by line into a destination file.

6. Write a C++ program to read an integer n, and display the nth line from the text file onto the screen.

7. Write a C++ program to print the number of lines, number of words, and number of characters in a given file.

8. To implement dictionaries using hashing with different hash functions.

9. To perform various operations on B-tree.
MSCS 1.7
Computer Organization Lab

Instruction: 3 Periods/week | Credits: 2
Internal: 50 Marks | University Exam: 50 Marks | Total: 100 Marks

I – CYCLE : Digital Logic Design Experiments

1. TTL Characteristics and TTL IC Gates
2. Multiplexers & Decoders
3. Flip-Flops
4. Counters
5. Shift Registers
6. Binary Adders & Subtractors
7. A L U

II – CYCLE: 8085 Assembly Language Programming :

1. 8085 Assembly Language Programming according to theory course microprocessors-I using the following trainers :
   - Keyboard Monitor of 8085µP Trainer.
   - Serial Monitor of 8085µP Trainer with Terminal
   - 8085 Line Assembler of 8085µP Trainer with PC as Terminal
   - 8085 Cross Assembler using In-Circuit Emulator (ICE) with 8085µP Trainer and PC as Terminal
   Graded Problems are to be used according to the syllabus of COMPUTER ORGANIZATION

2. PENTIUM CLASS PC ARCHITECTURE
   FAMILIARIZATION HARDWARE & SOFTWARE PARTS DEMONSTRATION
**Master of Science in Computer Science (M.Sc.)**

Course Structure and Scheme of Valuation w.e.f. 2016-17

### II SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>Name of the subject</th>
<th>Periods/week</th>
<th>Max. Marks</th>
<th>Total</th>
<th>Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
<td>Lab</td>
<td>Ext.</td>
<td>Int.</td>
</tr>
<tr>
<td>MSCS 2.1</td>
<td>Formal Languages &amp; Automata Theory</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.2</td>
<td>Relational Data Base Management Systems</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.3</td>
<td>Advanced Operating Systems</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.4</td>
<td>Elective I</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.5</td>
<td>Elective-II</td>
<td>4</td>
<td>--</td>
<td>70</td>
<td>30</td>
</tr>
<tr>
<td>MSCS 2.6</td>
<td>Advanced JAVA Programming Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>MSCS 2.7</td>
<td>Relational Data Base Management Systems Lab</td>
<td>--</td>
<td>3</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>20</td>
<td>6</td>
<td>450</td>
<td>250</td>
</tr>
</tbody>
</table>
MSCS 2.1  
**FORMAL LANGUAGES & AUTOMATA THEORY**

<table>
<thead>
<tr>
<th>Instruction: 3 Periods + 1 Tut/week</th>
<th>Credits: 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal: 30 Marks</td>
<td>University Exam: 70 Marks</td>
</tr>
</tbody>
</table>


3. Context Free Grammars and Languages: Context Free Grammars and Languages, Derivation Trees, Simplification of Context Free Grammars, Normal Forms, Pumping Lemma for CFL, Closure properties of CFL’s, Decision Algorithm for CFL.


5. Turing Machines: The Definition of Turing Machine, Design and Techniques for Construction of Turing Machines, Combining Turing Machines.

6. Universal Turing Machines and Undecidability: Universal Turing Machines. The Halting Problem, Variants of Turing Machines, Restricted Turing Machines, Decidable & Undecidable Problems - Post Correspondence Problem.

7. Chomsky Hierarchy of Languages: Regular Grammars, Unrestricted Grammars, Context Sensitive languages, Relationship between Classes of Languages.

**Text books:**

1. Introduction to Automata Theory, Languages and Computations – J.E. Hopcroft, & J.D. Ullman, Pearson Education Asia.

**Reference books:**

1. Introduction to languages and theory of computation – John C. Martin (MGH)
3. Introduction to Theory of Computation – Michael Sipser (Thomson Nrools/Cole)

2. **SQL**: Introduction to SQL Queries, Integrity Constraints, Joins, Views, Intermediate and Advanced SQL features and Triggers.


5. **Query Evaluation**: Overview, Query processing, Query optimization, Performance Tuning.


7. **Transaction Management**: Overview of Transaction Management, Transactions, Concurrency control, Recovery systems, Advanced Transaction Processing

8. **Case Studies**: Postgre SQL, Oracle, IBM DB2 Universal Database, Microsoft SQL Server.

**Text Books:**

**References:**
2. Schneider, Robert D & J. R. Garbus Optimizing SQL.


6. Distributed Systems & Synchronization: Clock Synchronization, Logical Clocks, Mutual Exclusion, Global Positioning Of Nodes, Data-Centric Consistency Models, Client-Centric Consistency Models, Consistency Protocols


8. Case Study: Over View Of UNIX, LINUX, Windows NT , Android And IOS Operating systems

Text Books:

References:


MSCS 2.4  
Elective I  
DESIGN AND ANALYSIS OF ALGORITHMS

Instruction: 3 Periods + 1 Tut/week  
Credits:4

Internal: 30 Marks  
University Exam: 70 Marks  
Total: 100 Marks

Course Objectives:  
On completing this course student will be able to

1. Analyze the asymptotic performance of algorithms.
2. Write rigorous correctness proofs for algorithms.
3. Demonstrate a familiarity with major algorithms and data structures.
4. Synthesize efficient algorithms in common engineering design situations.

Course Outcomes:

1. Students will be able to Argue the correctness of algorithms using inductive proofs and invariants and Analyze worst-case running times of algorithms using asymptotic analysis.
2. Describe the various paradigms of design when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm and synthesize them.
3. Students will be able to Compare between different data structures. Pick an appropriate data structure for a design situation.

Syllabus:


**Space and Time Tradeoffs** – Sorting by Counting – Input Enhancement in string Matching – Hashing – B-Trees

5. **Dynamic Programming** – Computing a Binomial Coefficient – Warshall’s and Floyd’s Algorithm – Optimal Binary Search Trees - The Knapsack Problem and MemoryFunctions


**Text Book:**

1. Introduction to Design & Analysis of Algorithms by Anany Levitin, Pearson Education, New Delhi, 2003

**Reference Books:**


2. **Image Transforms**: A Detail Discussion On Fourier Transform, DFT, FFT, Properties WALSH Transform, WFT, HADAMARD Transform, DCT

3. **Image Enhancement**: 
   a) Arithmetic and Logical Operations, Pixel or Point Operations, Size Operations,
   b) Smoothing Filters-Mean, Median, Mode Filters – Comparative Study
   c) Edge Enhancement Filters – Directorial Filters, Sobel, Laplacian, Robert, KIRSCH Homogeneity
   d) & DIFF Filters, Prewitt Filter, Contrast Based Edge Enhancement Techniques – Comparative Study
   e) Low Pass Filters, High Pass Filters, Sharpening Filters. – Comparative Study
   f) Colour Fundamentals and Colour Models
   g) Colour Image Processing.


5. **Image Compression**: Run Length Encoding, Contour Coding, Huffman Code, Compression Due to Change in Domain, Compression Due to Quantization Compression at the Time of Image Transmission. Brief Discussion on:- Image Compression Standards.

6. **Image Segmentation**: Characteristics of Segmentation, Detection of Discontinuities, Thresholding Pixel Based Segmentation Method. Region Based Segmentation Methods, Segmentation by Pixel Aggregation, Segmentation by Sub Region Aggregation, Histogram Based Segmentation, Spilt and Merge Technique, Motion in Segmentation

7. **Morphology**: Dilation, Erosion, Opening, Closing, Hit-And-Miss Transform, Boundary Extraction, Region Filling, Connected Components, Thinning, Thickening, Skeletons, Pruning Extensions to Gray – Scale Images Application of Morphology in I.P

8. **Image, Video & Multimedia Communications**: Multi-scale and multi-orientation representation; Geometry and texture representation; Object based representation; Hierarchical representation; Sparse representation, Multimedia with image and video content; Multimedia event synchronization;
Text Book:

1. Digital Image Processing, Rafael C. Gonzalez And Richard E. Woods, Addision Wesley
   Reference Books:
3. Image Processing, Analysis, And Machine Vision By Milan Sonka Vaclan Halavac Roger
   Boyle, Vikas Publishing House.


7. **Host and Target Machines** – Linker/Locator for Embedded Software- Getting Embedded Software into the Target System.


**Text Book:**


**Reference Book:**

MSCS 2.5 | Elective II | WEB TECHNOLOGIES

Instruction: 3 Periods + 1 Tut/week | Credits: 4

Internal: 30 Marks | University Exam: 70 Marks | Total: 100 Marks

1. Introduction to HTML, Core Elements, Links and Addressing, Images, Text, Colors and Background, Lists, Tables and Layouts, Frames, Forms, Cascading Style Sheets.

2. Introduction to Java Scripts, Elements of Objects in Java Script, Dynamic HTML with Java Script.


4. JDBC OBJECTS- JDBC Driver Types, JDBC Packages, Database Connection, Statement Objects, Result Set.

5. JDBC and Embedded SQL - Tables, Inserting Data into Tables, Selecting Data from a Table, Meta Data, Updating Table, Deleting data from Table, Joining Table, Calculating Data, Grouping and Ordering Data, Sub-queries, View.


8. Introduction to Java Beans, Using JAVA Bean Components in JSP Documents, MVC Architecture.

Text Books:
2. The complete Reference HTML and DHTML, Thomas A. Powey
3. The complete Reference J2ME, James Keogh

Reference Books:
1. Internet, World Wide Web, How to program, Dietel, Nieto, PHI/PEA
2. Web Tehnologies, Godbole, kahate, 2nd Ed., TMH


3. **WLAN Topologies**, WLAN Standard IEEE 802.11, Comparison Of IEEE 802.11a, B, G and N Standards, Wireless PANs, Hiper LAN, Wireless Local Loop, ATM, Virtual Private Networks, Wireless Data Services, Common Channel Signaling, Various Networks for Connecting to The Internet.


6. **Communications** Asymmetry, Classification of New Data Delivery Mechanisms, Push-Based Mechanisms, Pull-Based Mechanisms, Hybrid Mechanisms, Selective Tuning (Indexing) Techniques. CDMA, GSM, Wireless Data, 3GNetworks and Applications


**Text Books:**


**Reference Books:**

1. **Introduction**: The vision of Ambient Intelligence, Application examples, Types of applications, Challenges for Wireless Sensor Networks (WSNs), Sensor networks vs Enabling Technologies for WSNs, **Single node architecture**: Hardware components, Energy consumption of sensor nodes, Some examples of sensor nodes, Operating systems and execution environments

2. **Network architecture**: Sensor network scenarios, Optimization goals and figures of merit, Design principles for WSNs,

3. **Physical layer and transceiver design considerations in WSNs**: 
   **MAC Protocols**: Fundamentals of (wireless) MAC protocols, Low duty cycle protocols and wakeup concepts, Contention-based protocols, Schedule-based protocols, The IEEE 802.15.4 MAC protocol, How about IEEE 802.11 and Bluetooth

4. **Link layer protocols**: Fundamentals: tasks and requirements, Error control, Framing, Link management

5. **Naming and addressing**: Fundamentals, Address and name management in wireless sensor networks, Assignment of MAC addresses, Content-based and geographic addressing


7. **Data-centric and content-based networking** : Introduction, Data-centric routing, Data aggregation, Data-centric storage

8. **Transport layer and Quality of Service**: The transport layer and QoS in wireless sensor networks, Coverage and deployment, Reliable data transport, Single packet delivery, Block delivery, Congestion control and rate control

**TEXT BOOKS:**


Part-I

1. Java Program to Demonstrate Basic Concepts in Java
2. Program to demonstrate Multi threading using priorities
3. Program to demonstrate to demonstrate Files & string manipulators
4. Program to demonstrate Applet Program using Various Controls and perform Font animation
5. Program to demonstrate Menus, sub Menus, Popup Menus, Shortcut Keys, Check Boxes and Separators
6. Two Programs to demonstrate JDBC
7. Two Programs to demonstrate Servlets
8. Two Programs to demonstrate JSP
9. Two Programs to demonstrate Java Beans
10. Two Programs to demonstrate RMI

Part II Web Programming Lab Experiments

1. Design of the Web pages using various features of HTML and DHTML
2. Client server programming using servlets, ASP and JSP on the server side and java script on the client side
3. Web enabling of databases
4. Multimedia effects on web pages design using Flash.

Reference Books:

1. Internet and Web Technologies by Raj Kamal, TataMcGraw-Hill
Relational Data Base Management Systems Lab

Instruction: 3 Periods/week Credits: 2
Internal: 50 Marks University Exam: 50 Marks Total: 100 Marks

1. **Accessing the Database**: The first laboratory exercise is to connect to a database, populate it with data, and run very simple SQL queries. (Data Definition, Table Creation, Constraints, Insert, Select Commands, Update & Delete Commands.)
2. **Basic SQL**: This lab covers simple SQL queries. (Inbuilt functions in RDBMS.)
3. **Intermediate SQL**: This lab covers more complex SQL queries. (Nested Queries & Join Queries, Control structures)
4. **Advanced SQL**: This lab covers even more complex SQL queries. (Procedures and Functions, .PL/SQL, Cursors and Triggers)
5. **Database Access from a Programming Language**: This lab introduces you to database access from a programming language such as Java or C#. Although phrased using Java/JDBC, the exercise can be done using other languages, OBDC or ADO.NET APIs.
6. **Building Web Applications**: This lab introduces you to construction of Web applications. Although phrased using the Java Servlet API, the exercise can be done using other languages such as C# or PHP.
7. **Project**: Each student is assigned with a problem. The student is to develop a logical and physical database design for the problem and develop Forms, Menu design and Reports.

   A. The logical design performs the following tasks:
      1. Map the ER/EER diagrams to a relational schema. Be sure to underline all primary keys, include all necessary foreign keys and indicate referential integrity constraints.
      2. Identify the functional dependencies in each relation
      3. Normalize to the highest normal form possible

   B. Perform physical design based above logical design using Oracle/MSSQL on Windows platform and MySQL/PostgreSQL on Linux platform.

**Sample Term Projects:**

1. Retailer database
2. Automobile sales database
3. Electronics vendor database
4. Package delivery database
5. Real estate database

**References:**

2) ORACLE PL/SQL by example. Benjamin Rosenzweig, Elena Silvestrova, Pearson Education 3rd Edition
3) ORACLE Database Log PL/SQL Programming Scott Urman. TMG Hill.
4) SQL & PL/SQL for Oracle 10g, Black Book, Dr. P. S. Deshpande. Oracle PL/SQL Programming, Steven Feuerstein, O’Reilly Publishers.
### III SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>Name of the subject</th>
<th>Periods/week</th>
<th>Max. Marks</th>
<th>Total Credits</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSCS 3.1</td>
<td>Data Warehousing &amp; Mining</td>
<td>4</td>
<td>70/30</td>
<td>100/30</td>
</tr>
<tr>
<td>MSCS 3.2</td>
<td>Object Oriented Software Engineering</td>
<td>4</td>
<td>70/30</td>
<td>100/30</td>
</tr>
<tr>
<td>MSCS 3.3</td>
<td>Network Security &amp; Cryptography</td>
<td>4</td>
<td>70/30</td>
<td>100/30</td>
</tr>
<tr>
<td>MSCS 3.4</td>
<td>Elective III</td>
<td>4</td>
<td>70/30</td>
<td>100/30</td>
</tr>
<tr>
<td>MSCS 3.5</td>
<td>MOOCS-I</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>MSCS 3.6</td>
<td>OOSE Lab</td>
<td>--</td>
<td>50/50</td>
<td>100/20</td>
</tr>
<tr>
<td>MSCS 3.7</td>
<td>Network Programming &amp; Web Programming Lab</td>
<td>--</td>
<td>50/50</td>
<td>100/20</td>
</tr>
<tr>
<td>MSCS 3.8</td>
<td>Seminar on Advanced Topics</td>
<td>--</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td><strong>16</strong></td>
<td><strong>450</strong></td>
<td><strong>250</strong></td>
</tr>
</tbody>
</table>

**Elective III: Artificial Intelligence / Operations Research / Cloud computing/ BigData Analytics**

**MOOCS-I:**

Each student should learn any one of the following topics by registering for courses through Online instruction from standard e-learning portals like nptel, coursera, etc. and write the examination conducted as per the university norms.

**List of topics for MOOCS-I:**

- Internet of Things/ R Programming for Data Analytics/ Data Visualization using Tableau/
  MongoDB for Developers/ DevOps/Agile Technologies for Software Development
Data Warehousing & Data Mining

1. Introduction to Data Mining: Motivation and importance of Data Warehousing and Data Mining, Kinds of Data, Patterns, Technologies used, Major Issues in Data Mining, Data Mining Applications.

2. Understanding Data: Data Objects and Attributes Types, Statistical Descriptions of Data, Data Visualization, Estimating Data Similarity and Dissimilarity.

3. Data Warehouse and OLAP Technology: Basic Concepts of Data warehouse, Data Modeling using Cubes and OLAP, DWH Design and usage, Implementation using Data Cubes and indexing, types of OLAP servers, Data Generalization with Attribute Oriented Induction.

4. Data Preprocessing & Data Cube Technologies: need for Pre-processing the Data, Data Cleaning, Data Integration, Data Reduction, Data Transformation, Discretization and Concept Hierarchy Generation; Preliminary Concepts of Data Cube Computation, Data Cube Computation by Multi-way Array Aggregation for Full Cube

5. Mining Frequent Patterns Based on Associations and Correlations: Basic Concepts, Frequent Itemset Mining Methods: Apriori Algorithm, Association Rule Generation, Improvements to A Priori, FP-Growth Approach, Mining Frequent Patterns using Vertical Data Formats, Mining Closed and Max Patterns, Pattern Evaluation Methods

Text Book:
1. Data Mining- Concepts and Techniques by Jiawei Han, Micheline Kamber and Jian Pei – Morgan Kaufmann publishers – 3rd edition

References:
1. Data Mining Techniques, A.K.Pujari, University Press Data mining concepts by Tan, Steinbech, and Vipin Kumar - Pearson Edu publishers
2. Data Mining –Introductory and Advanced by Margarett Dunham -- Pearson Education publishers
3. Data Warehousing for Real –world by Sam Annahory-- Pearson Education publishers
1. **Introduction to Object Oriented Software Engineering**

2. **Requirements Engineering:** Domain Analysis, Problem Definition And Scope, Requirements Definition, Types Of Requirements, Techniques For Gathering And Analyzing Requirements, Requirement Documents, Reviewing, Managing Change In Requirements.


5. **Software Design And Architecture**

**Text Books:**
1. Object-Oriented Software Engineering Practical software development using UML and Java by Timothy C. Lethbridge & Robert, Langaniere Mcgraw-Hill

**Reference Books:**
1. **Introduction:** Introduction to Security, Security Approaches, Principles of Security; Security Services and Mechanism—confidentiality, Confidentiality, Authentication, Integrity, Non-repudiation, access Control and Availability; Conventional Encryption Principles, Conventional encryption algorithms, cipher block modes of operation, location of encryption devices, key distribution Approaches of Message Authentication, Secure Hash Functions and HMAC.


4. **Public Key Infrastructure:** Public key cryptography principles and algorithms, digital signatures, digital Certificates, Certificate Authority and key management, Public Key Cryptography Standards, Private Key Management, The PRIX Model, XML, PKI and Security.

5. **Symmetric Key Cryptographic Algorithms:** Overview of symmetric Key Cryptography Algorithm types and modes; DES, IDEA, RC5, BLOWFISH, AES Algorithms; Differential and Linear Cryptanalysis.

6. **Asymmetric Key Cryptographic Algorithms:** Overview of Asymmetric Key cryptography, RSA Algorithm, Symmetric and Asymmetric Key Cryptography Together, Digital Signature, Knapsack Algorithm and other Algorithms.

**Text Book:**

**Reference Books:**

1. Hack Proofing your network, Ryan Russell, Dan Kaminsky, Rain Forest Puppy, Joe Grand, David Ahmad, Hal Flynn Ido Dubrawsky, Steve W. Manzuik, Ryan Permeh, Wiley Dreamtech
7. Introduction to Cryptography, Buchmann, Springer
1. **Introduction to Artificial Intelligence**: Artificial Intelligence, AI Problems, AI Techniques, The Level of the Model, Criteria For Success. Defining the Problem as a State Space Search, Problem Characteristics, Production Systems, Production System Characteristics


3. **Knowledge Representation**: Procedural Vs Declarative Knowledge, Representations and Mappings, Approaches to Knowledge Representation, Issues in Knowledge Representation, Logic Programming Forward Vs Backward Reasoning,


**Text Books:**

Artificial Intelligence, Elaine Rich, Mcgraw-Hill Publications

**Reference Books:**

1. Introduction To Artificial Intelligence & Expert Systems, Patterson, PHI
2. Artificial Intelligence, George F Luger, Pearson Education Publications
3. Artificial Intelligence, Robert Schalkoff, Mcgraw-Hill Publications
1. Overview of Operations Research, Types of OR Models, Phases of Operations Research—OR Techniques, Introduction to Linear Programming, Formulation of Linear Programming Problem, Graphical Solution; Graphical Sensitivity Analysis,

2. Standard Form of LPP, Basic Feasible Solutions, Unrestricted Variables, Simplex Algorithm, Artificial Variables, Big M Method, Degeneracy, Alternative Optimal, Unbounded Solutions, Infeasible Solutions, Primal And Dual Problems And Their Relations, Dual Simplex Method

3. Transportation Problem as LPP, Initial Solutions, North West Corner Rule, Lowest Cost Method, Vogels Approximation Method, Optimum Solutions of TPP, Degeneracy in Transportation, Transportation Algorithms,


Text Books:

Reference Books:
2. Operations Research By S.D Sharma
4. Operations Research, Richard Bronson, Schaum’s Series, McGrawhill


4. **Software as a Service** - Overview, Driving Forces, Company Offerings, Industries Software plus Services - Overview, Mobile Device Integration, Providers, Microsoft Online.


**Text Books:**

MSCS 3.4  Elective-III  BIGDATA ANALYTICS

Instruction: 3 Periods + 1 Tut/week  Credits:4

Internal: 30 Marks  University Exam: 70 Marks  Total: 100 Marks

Course Objectives:

On completing this course student will be able to
1. Understand big data and Apache Hadoop Eco system
2. Understand distributed, parallel, cloud computing and SQL concepts
3. Apply Hadoop concepts
4. Understand concepts of map and reduce and functional programming

Course Outcomes:

1. Gain conceptual understanding of analytics concepts, algorithms and statistical tests
2. Students will be able to look at the core projects used for both batch and real time data processing such as Hadoop
3. Students will be able to look at wider range of problems and data science based solutions

1. Introduction to Big Data: Big Data-definition, Characteristics of Big Data (Volume, Variety, Velocity, Veracity, Validity), Importance of Big Data, Patterns for Big Data Development, Data in the Warehouse and Data in Hadoop.

2. Introduction to Hadoop: Hadoop-definition, Understanding distributed systems and Hadoop, Comparing SQL databases and Hadoop, Understanding MapReduce, Counting words with Hadoop—running your first program, History of Hadoop, Starting Hadoop - The building blocks of Hadoop, NameNode, DataNode, Secondary NameNode, JobTracker and Task Tracker

3. MapReduce -A Weather Dataset, Analyzing the Data with Unix Tools, Analyzing the Data with Hadoop, Scaling Out, Hadoop Streaming, Hadoop Pipes, Developing a MapReduce Application - The Configuration API, Configuring the Development Environment, Running Locally on Test Data, Running on a Cluster, Tuning a Job, MapReduce Workflows

4. HDFS: Components of Hadoop -Working with files in HDFS, Anatomy of a MapReduce program, Reading and writing the Hadoop Distributed File system -The Design of HDFS, HDFS Concepts, The Command-Line Interface, Hadoop Filesystem, The Java Interface, Data Flow, Parallel Copying with distcp, Hadoop Archives

5. MapReduce Programming: Writing basic Map Reduce programs - Getting the patent data set, constructing the basic template of a Map Reduce program, Counting things, Adapting for Hadoop’s API changes, Streaming in Hadoop, Improving performance with combiners.

Textbooks:

1. Dirk deRoos, Chris Eaton, George Lapis, Paul Zikopoulos, Tom Deutsch,


Reference Books:

1. Hadoop in Action by Chuck Lam, MANNING Publ.
2. Hadoop in Practice by Alex Holmes, MANNING Publishers
1. The purpose of the Software Engineering Lab course is to familiarize the students with modern software engineering methods and tools, Rational Products. The course is realized as a project-like assignment that can, in principle, by a team of three/four students working full time. Typically the assignments have been completed during the semester requiring approximately 60-80 hours from each project team.

2. The goal of the Software Engineering Project is to have a walk through from the requirements, design to implementing and testing. An emphasis is put on proper documentation. Extensive hardware expertise is not necessary, so proportionate attention can be given to the design methodology.

3. Despite its apparent simplicity, the problem allows plenty of alternative solutions and should be a motivating and educating exercise. Demonstration of a properly functioning system and sufficient documentation is proof of a completed assignment.

4. Term projects are projects that a group student or might take through from initial specification to implementation. The project deliverables include.

Projects:

1. Documentation including
   A. A problem statement
   B. A requirements document
      c. A Software Requirements Specification.

2. A design document
   A. A Software Design Description and a System Design Document.


4. Manuals/guides for
   A. Users and associated help frames
   B. Programmers
   C. Administrators (installation instructions)

5. A project plan and schedule setting out milestones, resource usage and estimated costs.
6. A quality plan setting out quality assurance procedures
7. An implementation.

References:
1. Project-based software engineering: An Object-oriented approach, Evelyn Stiller, Cathie LeBlanc, Pearson Education
3. UML2 Toolkit, Hans -Erik Eriksson, etc; Wiley
Part I Networks Lab Experiments

1. Identifying well known ports on a Remote System: By trying to listen to the various well known ports by opening client connections. If the exception does not occur then the remote port is active else the remote port is inactive.

2. Writing a Chat application:
   i. One-One: By opening socket connection and displaying what is written by one party to the other.
   ii. Many-Many (Broadcast): Each client opens a socket connection to the chat server and writes to the socket. Whatever is written by one party can be seen by all other parties.

3. Data retrieval from a Remote database: At the remote database a server listens for client connections. This server accepts SQL queries from the client, executes it on the database and sends the response to the client.

4. Mail Client:
   i. POP Client: Gives the server name, user name and password retrieve the mails and allow manipulation of mail box using POP commands.
   ii. SMTP Client: Gives the server name, send e-mail to the recipient using SMTP commands.

5. Simulation of Telnet: Provide a user interface to contact well-known ports, so that client-server interaction can be seen by the user.

6. Simple file transfer between two systems (without protocols): By opening socket connection to our server on one system and sending a file from one system to another.

7. TFTP-Client: To develop a TFTP client for file transfer. (Unix Network Programming-Stevens.)

8. HTTP-Server: Develop a HTTP server to implement the following commands. GET, POST, HEAD, DELETE. The server must handle multiple clients.

Reference Books:

2. An Introduction to Computer Networking, Kenneth C. Mansfield Jr and James L. Antonakos, Pearson Education Asia
Purpose:

1. To enable a student to be familiar with Communication skills
2. Student is expected to Learn

a. How to Make a Presentation
   i. Verbal
   ii. Non Verbal
   iii. LCD based Power Point

b. How to write a report
   i. Abstract
   ii. Body
   iii. Conclusions
   iv. Executive Summary

c. Group Discussion
   i. Share the work with a group
   ii. Modularization of the work
   iii. Shareware Development

d. Communication
   i. Horizontal
   ii. Vertical

Students will be Given a Topic Of Importance and are Expected

A. To Present the Topic Verbally in 45 minutes + Question Answering
B. To Present the Topic as a Report in 50 Pages
Master of Science in Computer Science (M.Sc.)
Course Structure and Scheme of Valuation w.e.f. 2016-17

IV SEMESTER

<table>
<thead>
<tr>
<th>Code</th>
<th>NAME OF THE SUBJECT</th>
<th>MAXIMUM MARKS</th>
<th>CREDITS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>INTERNAL</td>
<td>EXTERNAL</td>
</tr>
<tr>
<td>MSCS 4.1</td>
<td>PROJECT/ THESIS WORK</td>
<td>50</td>
<td>50</td>
</tr>
</tbody>
</table>

Total Credits (Complete Course) 87

Instructions for Project

1. Three Stages In Project adjudication:

   Stage I: Presentation of Concept Note & Problem Approval by Guide
   Stage II: Progress Approval by System Demonstration with results Internal -50 Marks
   Stage III: Final Presentation with Documentation & External Viva-Voce - 50 Marks

2. Candidates can do their thesis work within the department or in any industry/research organization for two semesters (4th semesters). In case of thesis done in an industry/research organization, one advisor (Guide) should be from the department and one advisor (CO-Guide) should be from the industry/research organization.

3. A publication of a paper on the thesis work in a National/International Conference proceedings with presentation certificate or a paper on the thesis work be communicated to a National/International Journal & accepted for publication for the submission of thesis at the end of 4th semester is desirable.

4. The external examiner shall be nominated by the Chairman, Board of Examiners in CSSE as per the norms of the University.