REGULATIONS - 2013

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING

CURRICULUM AND SYLLABI OF
M.C.A – MASTER OF COMPUTER APPLICATIONS

I & II YEAR
REGULATIONS - 2013
Curriculum and Syllabi of Full Time
M.C.A. – Master of Computer Applications

SEMMESTER - I

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>MCC11</td>
<td>Computer Organization</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>MCC12</td>
<td>Problem Solving and C Programming</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>MCC13</td>
<td>Database Management Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>MCC14</td>
<td>Data Structures</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>MCC15</td>
<td>Mathematical Foundations of Computer Science</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Practical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>MCC16</td>
<td>C and Data Structures Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7.</td>
<td>MCC17</td>
<td>Database Management Systems Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOTAL</td>
<td>15</td>
<td>1</td>
<td>6</td>
<td>20</td>
</tr>
</tbody>
</table>

SEMMESTER – II

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>MCC21</td>
<td>Object Oriented Programming</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>2.</td>
<td>MCC22</td>
<td>Design and Analysis of Algorithms</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>3.</td>
<td>MCC23</td>
<td>System Software</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>MCC24</td>
<td>Operating Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>MCC25</td>
<td>Software Engineering</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Practical</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>MCC26</td>
<td>Object Oriented Programming Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7.</td>
<td>MCC27</td>
<td>System Programming Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>MCC28</td>
<td>Communication Skills Laboratory</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TOTAL</td>
<td>16</td>
<td>2</td>
<td>8</td>
<td>23</td>
</tr>
</tbody>
</table>
SEMESTER – III

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>MCC31</td>
<td>Java Programming</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>MCC32</td>
<td>Object Oriented Analysis and Design</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>MCC33</td>
<td>Computer Graphics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>MCC34</td>
<td>Computer Networks</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>MCC35</td>
<td>Resource Management Techniques</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Practical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>MCC36</td>
<td>Java Programming Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7.</td>
<td>MCC37</td>
<td>Case Tools Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>MCC38</td>
<td>Graphics Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>15</td>
<td>0</td>
<td>9</td>
<td>21</td>
</tr>
</tbody>
</table>

SEMESTER – IV

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>MCC41</td>
<td>Network Programming</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>MCC42</td>
<td>Web Programming</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>MCC43</td>
<td>Compiler Design</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>Elective – I</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>Elective – II</td>
<td></td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>Practical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>MCC44</td>
<td>Network Programming Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7.</td>
<td>MCC45</td>
<td>Web Programming Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>MCC46</td>
<td>Compiler Design Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>15</td>
<td>0</td>
<td>9</td>
<td>21</td>
</tr>
</tbody>
</table>
## SEMESTER – V

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>MCC51</td>
<td>.NET Programming and Scripts</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>MCC52</td>
<td>XML and Web Services</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td></td>
<td>Elective – III</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td></td>
<td>Elective – IV</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td></td>
<td>Elective – V</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Practical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>MCC53</td>
<td>XML and Web Services Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>7.</td>
<td>MCC54</td>
<td>.NET Programming Laboratory</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>8.</td>
<td>MCC55</td>
<td>Mini Project Work</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td>15</td>
<td>0</td>
<td>9</td>
<td>21</td>
</tr>
</tbody>
</table>

## SEMESTER – VI

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Practical</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.</td>
<td>MCC61</td>
<td>Project Work</td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>TOTAL</td>
<td></td>
<td>0</td>
<td>0</td>
<td>24</td>
<td>12</td>
</tr>
</tbody>
</table>

TOTAL CREDITS TO BE EARNED FOR THE AWARD OF THE DEGREE - 118
LIST OF ELECTIVES OF IV SEMESTER

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MCE4A</td>
<td>Numerical and Statistical Methods</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>MCE4B</td>
<td>Electronic Commerce</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>MCE4C</td>
<td>Information Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>MCE4D</td>
<td>Web Graphics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>MCE4E</td>
<td>Advanced Databases</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>MCE4F</td>
<td>Software Quality Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>7.</td>
<td>MCE4G</td>
<td>TCP/IP Design and Implementation</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8.</td>
<td>MCE4H</td>
<td>Distributed Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>9.</td>
<td>MCE4J</td>
<td>Unix Internals</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>10.</td>
<td>MCE4K</td>
<td>Visual Programming</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>

LIST OF ELECTIVES OF V SEMESTER

<table>
<thead>
<tr>
<th>Sl. No.</th>
<th>Course Code</th>
<th>Course Title</th>
<th>L</th>
<th>T</th>
<th>P</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>MCE5A</td>
<td>Human Resource Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2.</td>
<td>MCE5B</td>
<td>Data Mining and Data Warehousing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>3.</td>
<td>MCE5C</td>
<td>Component Based Technology</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>4.</td>
<td>MCE5D</td>
<td>Managerial Economics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>5.</td>
<td>MCE5E</td>
<td>Mobile Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>6.</td>
<td>MCE5F</td>
<td>Digital Imaging</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>7.</td>
<td>MCE5G</td>
<td>Enterprise Resource Planning</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>8.</td>
<td>MCE5H</td>
<td>Agent Based Intelligent Systems</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>9.</td>
<td>MCE5J</td>
<td>Natural Language Processing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>10.</td>
<td>MCE5K</td>
<td>Software Agents</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>11.</td>
<td>MCE5L</td>
<td>Supply Chain Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>12.</td>
<td>MCE5M</td>
<td>Healthcare for IT Services</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>13.</td>
<td>MCE5N</td>
<td>Portfolio Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>14.</td>
<td>MCE5P</td>
<td>Artificial Intelligence</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>15.</td>
<td>MCE5Q</td>
<td>Parallel and Distributed Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>16.</td>
<td>MCE5R</td>
<td>Soft Computing</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>17.</td>
<td>MCE5S</td>
<td>Software Project Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>18.</td>
<td>MCE5T</td>
<td>Professional Ethics</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>19.</td>
<td>MCE5V</td>
<td>Mobile Engineering</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>20.</td>
<td>MCE5W</td>
<td>Infrastructure Administration and Management</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
</tbody>
</table>
MCC11 COMPUTER ORGANIZATION

OBJECTIVES
- To study the Digital fundamentals.
- To understand the concepts and issues of Computer Organization.
- To learn the basics of Memory and Processor design.
- To describe the Interrupts and I/O interfaces.

UNIT I DIGITAL FUNDAMENTALS

UNIT II COMBINATIONAL AND SEQUENTIAL CIRCUITS

UNIT III BASIC STRUCTURE OF COMPUTERS
Functional units – Basic operational concepts – Bus structures – Performance and Metrics – Instruction and Instruction sequencing – Addressing modes – Instruction Set: ARM Processor – ALU design – Fixed point and Floating point operation.

UNIT IV PROCESSOR DESIGN

UNIT V MEMORY AND I/O SYSTEM

TOTAL: 45

TEXT BOOKS

REFERENCES
MCC12 PROBLEM SOLVING AND C PROGRAMMING L T P C 3 0 0 3

OBJECTIVES
- To learn and analyze problems and formulate algorithms.
- To learn the fundamentals of C.
- To understand the usage of arrays, functions and structures.
- To study the importance of pointers and files.

UNIT I INTRODUCTION TO PROGRAMMING 9

UNIT II C PROGRAMMING BASICS 9

UNIT III ARRAYS AND FUNCTIONS 9
Arrays – One dimensional and two dimensional arrays - String – String operations – String Arrays – Functions – Call by value – Call by reference – Recursion.

UNIT IV STRUCTURES AND UNIONS 9

UNIT V POINTERS AND FILE MANIPULATION 9

TOTAL: 45

TEXT BOOKS

REFERENCES
MCC13 DATABASE MANAGEMENT SYSTEMS  

**OBJECTIVES**
- To understand the basics of Database Management System.
- To use Standard Query Language.
- To understand normalization techniques.
- To understand importance of backup and recovery techniques.

**UNIT I** INTRODUCTION  

**UNIT II** SQL, PROGRAMMING AND TRIGGERS  

**UNIT III** NORMAL FORMS, STORAGE AND FILE STRUCTURE  

**UNIT IV** HASHING, QUERY PROCESSING AND TRANSACTION  

**UNIT V** CONCURRENCY CONTROL AND RECOVERY SYSTEM  

**TOTAL**: 45

**TEXT BOOK**

**REFERENCES**
OBJECTIVES

- To learn about the basic linear data structures, their implementations and applications.
- To make the students to study about non linear data structures.
- To learn about various sorting types and its efficiency.
- To make the students to learn about hashing techniques.

UNIT I       LINEAR DATA STRUCTURES  9
Elementary Data Structures – ADT – List ADT: Array based Implementation – Doubly and Circularly
Linked Lists – Cursor based Linked List – Applications of List – The stack ADT: Model –

UNIT II      TREE STRUCTURES AND BALANCED TREES  9
Trees - Preliminaries – implementation of Tree – Tree Traversals – Binary tree Implementation –
Expression Tree – The Search Tree ADT – Binary Search Tree Operations – AVL tree – Splay tree –

UNIT III     GRAPH  9
Graph - Representation of Graphs – Bi-connectivity – Topological sort – Graph Traversal:
Breadth-First Traversal – Depth-First Traversal – Shortest path Algorithms: Minimum Spanning Tree
– Prim's and Kruskal's Algorithms – Applications of Graph.

UNIT IV      SORTING  9
Sorting – Lower Bound for Simple Sorting Algorithms – Internal and External Sorting: Bubble Sort –
Insertion Sort – Shell Sort – Merge Sort – Heap Sort – Quick Sort – Bucket Sort – Radix Sort –
Distribution Sort – Shuffle Sort.

UNIT V       HASHING AND SET  9
Hashing: Hash function – Open hashing – Closed hashing – Double hashing – Extendible hashing -
Rehashing. The Disjoint set ADT: Equivalence Relations – Dynamic Equivalence Problem – Smart
Union Algorithms – Path compression – Applications of Set.

TOTAL: 45

TEXT BOOK

REFERENCES
MCC15  MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE  L T P C
                                                3 1 0 4

OBJECTIVES
• To learn about the basic linear algebra concepts.
• To understand the relations and logics of computer science.
• To make the students to understand the basics of network theory using graph.

UNIT I  MATRIX ALGEBRA  12

UNIT II  SETS, RELATIONS AND FUNCTIONS  12

UNIT III  MATHEMATICAL LOGIC  12

UNIT IV  FORMAL LANGUAGES  12

UNIT V  GRAPH THEORY  12
Graph – Special types of graphs – Sub graph – Graph isomorphism – Euler graph – Hamiltonian graph.

TOTAL: 60

TEXT BOOKS

REFERENCES
MCC16  C AND DATA STRUCTURES LABORATORY  L T P C  0 0 3 2

OBJECTIVES

- To implement various operations on linear data structures.
- To perform the various tree traversals in binary tree.
- To implement various internal and external sorting techniques.
- To implement shortest path algorithms using Floyd’s algorithm and Warshall’s algorithm.
- To implement minimum spanning tree of graph using Prim’s algorithm and Kruskal’s algorithm.

List of Experiments:

1. Write a program to perform the following operations on single linked list:
   i) Creation ii) Insertion iii) Deletion iv) Reversal v) Sorting vi) Traversal

2. a) Write a program that uses functions to perform the following operations on Double linked list:
   i) Creation ii) Insertion iii) Deletion iv) Traversal
   b) Write a program to create and traverse a circular single linked list

3. a) Write a program to create a stack using array and linked list and perform the following operations:
   i) push ii) pop iii) display
   b) Write a program to create a queue using array and linked list and perform the following operations:
   i) insert ii) delete iii) display

4. Write a program to perform the following stack applications:
   i) Convert infix expression into postfix expression
   ii) Evaluation of postfix expression

5. Write the program to perform the following traversal in Binary tree
   i) Inorder traversal ii) Preorder traversal iii) Postorder traversal

6. Write a program to perform the following operations in Binary Search Tree (BST):
   i) Creation ii) Insertion iii) Deletion iv) Traversal

7. Write a program that implement the following sorting techniques:
   i) Bubble sort ii) Insertion sort iii) Selection sort

8. Write a program that implement the following sorting techniques:
   i) Quick sort ii) Merge sort iii) Heap sort

9. Write a program to perform following Traversal in Graph
   i) Breadth First Search, ii) Depth first search

10. Write a program to find out the shortest path between every vertex in Graph using
    the following algorithms
    i)Floyd’s algorithm ii)Warshall’s algorithm

11. Write a program to find out the shortest path in Graph using the following algorithms
    i) Prim’s algorithm ii) Kruskal’s Algorithm

12. Write a program to implement Huffman’s Algorithm.

Required Software: C/C++
MCC17 DATABASE MANAGEMENT SYSTEMS LABORATORY L T P C
0 0 3 2

OBJECTIVES
- To implement the Basic Queries.
- To implement the SQL program.
- To implement the GUI program.

List of Experiments
1. Execute DDL commands for tables like Bank Management System.
2. Execute DML commands for table for Student Processing System.
3. Execute and manipulate the views for Bank Management System.
4. Apply different joins like equi-join and outer join by relating two or more tables in Payroll processing system.
5. Manipulate student processing system using SET operations and Aggregate functions.
6. Execute Built-in functions for Characters, Numbers and Date.
7. Write basic PL/SQL programming for the following:
   a) Write a programme to find the factorial of a number.
   b) Find odd and even numbers from 100 to 1000.
   c) Write a program to input 2 numbers if the 1st no > 2nd no then swap it, else if 1st no < 2nd no doubles it else multiply 10 with both numbers?
   d) Write a program to input 2 numbers and an operator, and display the result.
   e) Print multiplication table from 1 to a given Number.
8. Do Advanced PL/SQL programming using
   a) Cursors b) Procedures c) Triggers
9. Create Forms, Menus and Report for
   a) Payroll Processing System
   b) Bank Management System
   c) Library Management System

Required Software: Oracle, Visual Basic.
MCC21 OBJECT ORIENTED PROGRAMMING

OBJECTIVES
- To understand the fundamentals of OOP’s Concepts.
- To understand the C++ Concepts.
- To understand the file operations in C++.

UNIT I FUNDAMENTALS

UNIT II FUNCTIONS, CONSTRUCTORS AND DESTRUCTORS

UNIT III INHERITANCE AND POLYMORPHISM
Need of Inheritance - public, private, and protected derivations – Access control. Types of inheritance: Single inheritance, Multilevel inheritance, Multiple inheritance, Hierarchical inheritance and Hybrid inheritance – Virtual base classes – Constructors in derived class. Pointers – Pointers to Objects – this Pointer – Pointers to derived classes - Virtual and Pure Virtual Functions – Virtual Constructors and Destructors.

UNIT IV I/O FORMATTING AND FILE HANDLING

UNIT V TEMPLATES
Drawbacks of Macros - Function Templates – Function Templates with Multiple / Two Generic Arguments – Overloading of Template Function – Member Function Templates – Class Templates – Class Template with Multiple Arguments – Exception Handling.

TEXT BOOKS

REFERENCES
MCC22 DESIGN AND ANALYSIS OF ALGORITHMS L T P C

3 1 0 4

OBJECTIVES

- To study about the fundamentals of problem solving and algorithm analysis.
- To understand the problem using Divide and conquer methods and Greedy technique.
- To learn about Dynamic programming techniques to solve Knapsack problem.
- To study about N Queens problem, sum of subset problem using Backtracking method.
- To learn about approximation algorithm for NP-hard and NP-complete problems.

UNIT I INTRODUCTION


UNIT II DIVIDE AND CONQUER METHOD AND GREEDY METHOD


UNIT III DYNAMIC PROGRAMMING


UNIT IV BACKTRACKING AND BRANCH AND BOUND


UNIT V NP-HARD AND NP-COMPLETE PROBLEMS


L: 45  T: 15  TOTAL: 60

TEXT BOOKS


REFERENCES

### MCC23 SYSTEM SOFTWARE

<table>
<thead>
<tr>
<th>OBJECTIVES</th>
</tr>
</thead>
<tbody>
<tr>
<td>To discuss the machine architecture of SIC &amp; SIC/XE.</td>
</tr>
<tr>
<td>To elaborate the functionality of an assembler, loader and linker with necessary data structures.</td>
</tr>
<tr>
<td>To differentiate between machine dependent and machine independent assembler features.</td>
</tr>
<tr>
<td>To design assemblers, loaders, linkers and text editors.</td>
</tr>
<tr>
<td>To describe the functionality of macro processors.</td>
</tr>
</tbody>
</table>

#### UNIT I INTRODUCTION
9
Introduction – System software and machine architecture – The Simplified Instructional Computer (SIC) – Machine Architectures (SIC and SIC/XE) – Data and Instruction Formats – Addressing Modes – Instruction sets – I/O.

#### UNIT II ASSEMBLERS
9

#### UNIT III LOADERS AND LINKERS
9

#### UNIT IV MACRO PROCESSORS
9

#### UNIT V TEXT EDITOR AND DEBUGGER
9

**TOTAL: 45**

### TEXT BOOK

### REFERENCE
MCC24 OPERATING SYSTEMS                          L T P C
                                                   3 0 0 3

OBJECTIVES

• Understand how the operating system abstractions can be used in the development of application programs or to build higher level abstractions.
• Understand how the operating system abstractions can be implemented.
• Understand the principles of concurrency and synchronization, and apply them to write correct concurrent programs/software.
• Understand basic resource management techniques and principles and how they can be implemented. These also include issues of performance and fairness objectives, avoiding deadlocks, as well as security and protection.

UNIT I INTRODUCTION                                9

UNIT II PROCESS MANAGEMENT                        9

UNIT III PROCESS SYNCHRONIZATION AND DEADLOCKS    9

UNIT IV STORAGE MANAGEMENT                       9

UNIT V CASE STUDY                                 9

TOTAL: 45

TEXT BOOK

REFERENCES
MCC25 SOFTWARE ENGINEERING L T P C
3 0 0 3

OBJECTIVES
• To gain knowledge of basic software engineering methods and practices, and their appropriate application.
• A general understanding of software process models such as the waterfall and evolutionary models.
• An understanding of the role of project management software requirements SRS document, implementation issues, verification and validation reviews, software testing approaches, software evolution, version management and how to ensure good quality software.

UNIT I INTRODUCTION 9

UNIT II SOFTWARE REQUIREMENT SPECIFICATIONS (SRS) 9

UNIT III SOFTWARE DESIGN 9

UNIT IV SOFTWARE TESTING 9

UNIT V SOFTWARE MAINTENANCE 9

TOTAL: 45

TEXT BOOK:

REFERENCES:
MCC26 OBJECT ORIENTED PROGRAMMING LABORATORY L T P C 0 0 3 2

OBJECTIVES
- To implement the basic concepts of C++.
- To implement the polymorphism and template concepts.
- To implement the file operations.

List of Experiments:
1. Write a program to perform a student’s internal mark calculation.
2. Write a program to calculate the area of circle, rectangle and triangle using function overloading
3. Write a class Square which has a field for side. It must have a constructor to initialize the side. Add methods to the Square class to calculate area and perimeter.
4. Write a class Circle which has a field for radius. It must have a constructor to initialize the radius. Add methods to the Circle class to calculate area and perimeter.
5. Write a class CheckoutCalculator which behaves somewhat like the machine at the checkout counter in supermarkets. This calculator should ask for the number of items for which the total is to be calculated and then allow you to enter the price for every item. After entering all the items it displays all the prices entered and the total amount. (Hint: Here you need to use dynamic memory allocation since you do not know how many items will be there. Use an array to store the prices of items)
6. Write a class Results which stores all the results in an array. Assume that we need to store only the results of a single semester which is four results. Each result will be final marks for a course between 0-100 and is stored in an integer array. Use dynamic memory allocation and the four methods mentioned above. Add separate methods to calculate the total marks for all courses and the average marks. Add a field for student ID.
7. Modify the Circle class which you have written earlier to overload the + operator so that you can add two Circle objects. Adding two Circle object should give another Circle whose radius is the sum of the radius of the two Circle objects.
8. Modify the Rectangle class which you have written earlier to overload the + operator so that you can add two Rectangle objects. Adding two Rectangle objects should give another Rectangle object whose length is the sum of the lengths of the two Rectangle objects and whose breadth is the sum of the breadths of the two Rectangle objects.
9. Write a class Time which represents time. The class should have three fields for hours, minutes and seconds. It should have constructor to initialize the hours, minutes and seconds. A method printTime() to print the current time. Overload the following operators: plus operator (+) (add two time objects based on 24 hour clock) and < (compare two time objects)
10. Modify the Rectangle class which you have written earlier to overload the < (less than) operator so that you can compare two Rectangles as to which is bigger (or smaller). The logic you will use is to compare the areas of the Rectangle objects and decide which is bigger.

11. Modify the Circle class which you have written earlier to overload the > (greater than) operator so that you can compare two Circles as to which is bigger (or smaller). The logic you will use is to compare the radii of the Circle objects and decide which is bigger.

12. Write a program to perform Stack and Queue operations using template.

13. Write a function calculateAverage() which takes four int arguments which are marks for four courses in the semester and returns their average as a float. The calculateAverage() function should take only valid range for marks which is between 0 - 100. If the marks are out of range throw an OutOfRangeException - define this exception as a class.

14. Write a program to perform a bank management system using file.

Required Software: C++
MCC27      SYSTEM PROGRAMMING LABORATORY                             L T P C
                      0  0  3  2

OBJECTIVES

- To develop and analyze the data structures needed for developing an assembler, loader and linker.
- To perform file handling using file manipulation system calls
- To implement process management
- To implement IPC techniques
- To analyze scheduling algorithms and demonstrate page replacement policies

1. Write a C program to create symbol table and intermediate file using the algorithm for pass1 of the two pass assembler.
2. Write a C program to create object file and list file with the output of the previous program using the algorithm for pass2 of the two pass assembler.
3. Write a C program to implement the absolute loader.
4. Write a C program to implement relocating loader.
5. Write a C Program to implement the pass1 of the linking loader.
6. Write a C program to implement the pass2 of the linking loader.
7. Develop a text editor with features like insertion / deletion of a character, word, and sentence.
8. Write a C program to perform file management tasks using file manipulation system calls in UNIX creat( ), open( ), read( ), write( ), close( )
9. Write a C program to create a new process using fork( ). Make the child process to execute a new program using exec( ). Terminate the execution of the child process using exit( ). Make the parent to read the termination status of the child process using wait( ) / waitpid( ).
10. Write a C program to implement IPC using pipe( ).
11. Write a C program to implement IPC using message queue for unrelated process.
12. Write a program to implement the following process scheduling algorithms
    a. First Come First Serve
    b. Shortest Remaining Job First (preemptive & non preemptive)
    c. Round Robin
    d. Priority Scheduling
13. Write a program that demonstrates how two processes can share a variable using semaphore.
14. Write a program to implement producer consumer problem using semaphore.
15. Write a program to demonstrate page replacement policies like
    a. Optimal
    b. Least Recently Used (LRU)
    c. First-In-First-Out

Required Software: LINUX & C
MCC28 COMMUNICATION SKILLS LABORATORY

OBJECTIVES

• To help the learners to improve their communicative skill.
• To facilitate the learners to improve the pronunciation of words with proper stress.
• To help the learners acquire the soft skills and interpersonal skills which will help the student to excel in their workplace.
• To inculcate the habit of reading and to improve the active vocabulary among the learners.
• To enhance the performance of students in placement, interviews and Group discussion.

Unit I

1. Vocabulary Building
2. Splitting Syllables
3. Stress and Shift of words and sentences
4. Common errors in Speaking
5. Letter writing
7. Reading Comprehension and Answering Multiple Choice questions and Fill ups.

Unit II

1. Listening to audio files and answering questions
2. Planning for an event
3. Extempore Speech - On the spot topics for speech Practice
4. Identifying tonal variations expressing
5. E-Mail writing

Unit III

1. Listening to Presentation Skills (GD & Debate)
2. Group Discussion
3. Reading Practice: Dr. Abdul Kalam’s “Wings of Fire”
4. Report Writing
5. Paper Presentation

Unit IV

1. Listening to Telephonic Conversation & Situational Conversation
2. Debate
3. Note Taking
4. Interview Skills

RECORD LAY OUT

Every student has to maintain a record in which he / she have to incorporate the following details.

1. Students have to collect materials related to topics for Group Discussion / Debate
2. 10 assignments of Lab observations related to Presentation Skills about 200 words each
3. Covering letter with Bio data / Resume / Curriculum Vitae
4. Paper Presentation Topics with source materials to be pasted in the record

TOTAL: 45

Required Software: Oral Digital Language Lab & Globarena
MCC31  JAVA PROGRAMMING

L T P C
3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to

• Analyze and design a computer program to solve real world problems based on object-oriented principles of Java.
• Understand and able to design applications using multithreading and well suitable runtime error handling.
• Develop GUI interfaces for a computer program to interact with users, and to understand the event-based GUI handling principles.
• Develop applications with components and patterns of Networking and database activities.

UNIT I  INTRODUCTION  7

UNIT II  INHERITANCE  9

UNIT III  EXCEPTION AND MULTITHREADING  9

UNIT IV  APPLET AND AWT  12

UNIT V  I/O AND NETWORKING  8

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
MCC32 OBJECT ORIENTED ANALYSIS AND DESIGN

COURSE OUTCOMES
On successful completion of the course, the students will be able to
- Develop the knowledge of structural and behavioral modeling techniques.
- Develop effective model-based software development methodology.
- Implement the design patterns and their application in a software design project.
- Demonstrate knowledge of Design and Testing Process Improvement Models.

UNIT I REVIEW OF OBJECT ORIENTATION

UNIT II METHODOLOGY AND UML

UNIT III OBJECT ORIENTED ANALYSIS

UNIT IV OBJECT ORIENTED DESIGN

UNIT V IMPLEMENTATION, USABILITY, TESTING AND QUALITY

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
MCC33  COMPUTER GRAPHICS  L T P C  3 0 0 3

COURSE OUTCOMES

On successful completion of the course, the students will be able to

• Perform pixel-based processing to create simple geometric figures upon a screen.
• Implement transformations of graphical objects in two and three dimensions, project such objects from three to two dimensions and perform hidden-surface removal on faceted models.
• Understand the principles of color models and animation.
• Understand the use of fractal geometry and ray tracing to the problem of improving visual realism.
• Demonstrate practical competence in the use of graphics.

UNIT I  INTRODUCTION  9
I/O devices – I/O primitives – Attributes of output primitives – DDA – Bresenham technique – Circle drawing algorithms – Interactive input methods.

UNIT II  2D GRAPHICS  9

UNIT III  3D GRAPHICS  9

UNIT IV  GRAPHICS PROGRAMMING  9

UNIT V  FRACTALS  9
Fractals and Self similarity – Creating image by iterated functions – Mandelbrot sets – Random Fractals – Overview of Ray Tracing – Intersecting rays with other primitives – Adding Surface texture – Reflections and Transparency – Boolean operations on Objects.

TOTAL: 45 PERIODS

TEXT BOOKS


REFERENCES

MCC34            COMPUTER NETWORKS            L T P C
                          3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
  • Analyze and solve the data transmission medium oriented numerical problems.
  • Construct a network with different topologies.
  • Analyze the difference between reliable and unreliable secure data delivery service.

UNIT I   INTRODUCTION
Communication model – Data Communications – Data Transmission: analog and digital data
transmission, Transmission impairments, channel capacity – Transmission media: Guided
Control Protocols: Flow Control, Error Control, High-Level Data Link Control (HDLC).

UNIT II  NETWORK FUNDAMENTALS
Need for protocol architecture – Reference models: OSI Reference Model, TCP/IP Reference Model,
Comparison of the OSI and TCP/IP Reference Models – LAN Protocol Architecture: Topologies and
Transmission media, Medium Access Control (MAC) and Logical Link Control (LLC) – Bridges:
Functions of a Bridges, Bridge Protocol Architecture, Fixed Routing and Spanning Tree Approach.

UNIT III NETWORK LAYER
Routing Strategies, BGP and OSPF Protocol – Congestion control – Internet Protocol: IP Services,
IPv4, IP Addresses, Subnets, ICMP, ARP and RARP.

UNIT IV TRANSPORT LAYER
Connection-Oriented Transport Protocol Mechanism: Reliable Sequencing Network Service,
Unreliable Network Service – Transmission Control Protocol (TCP) – TCP Congestion Control:
Additive Increase/Multiplicative Decrease, Slow Start and, Fast Retransmit and Fast Recovery – User
Datagram Protocol (UDP).

UNIT V APPLICATIONS
Security: Requirements and Attacks, Message Authentication and Secure Hash Algorithm, RSA
Public Key Encryption and Digital Signature – DNS – SNMP – Electronic mail: Architecture and
Services, User agent, Message Format (MIME) and Message Transfer (SMTP, E-mail Gateway and
Final Delivery) – World Wide Web (WWW): HTTP and URLs.

TOTAL: 45 PERIODS

TEXT BOOKS
   Education Inc., 2011.

REFERENCES
1. Larry L. Peterson and Bruce S. Davie, “Computer Networks – A Systems Approach”, 5th
COURSE OUTCOMES

On successful completion of the course, the students will be able to
- Formulate real practical problems by mathematical modeling.
- Solve a LPP, transportation, assignment problems using resource management techniques.
- Apply scheduling and queuing models and propose solutions to them.
- Use computer tools to solve a mathematical model for a practical problem.

UNIT 1  LINEAR PROGRAMMING MODELS  9
Mathematical Formulation of Linear Programming problems – Graphical Solution method – Simplex method – slack, surplus and artificial variables, two phase method and Big-M method artificial variable Techniques – Variants of Simplex method.

UNIT II  TRANSPORTATION AND ASSIGNMENT MODELS  9

UNIT III  INTEGER PROGRAMMING MODELS  9
Formulation – Gomory’s IPP method – Gomory’s mixed integer method – Branch and bound technique.

UNIT IV  SCHEDULING BY PERT AND CPM  9

UNIT V  QUEUEING MODELS  9
Characteristics of Queuing Models – Poisson Queues – (M / M / 1) : (FIFO / ∞ / ∞), (M / M / 1) : (FIFO / N / ∞), (M / M / C) : (FIFO / ∞ / ∞), (M / M / C) : (FIFO / N / ∞) models.

TOTAL: 45 PERIODS

TEXT BOOK

REFERENCES
MCC36                JAVA PROGRAMMING LABORATORY                  L T P C
                        0 0 3  2

COURSE OUTCOMES
On successful completion of the course, the students will be able to

- Implement, compile, test and run Java program.
- Make use of hierarchy of Java classes to provide a solution to a given set of requirements found in the Java API.
- Understand the components and patterns that constitute a suitable development for GUI application and event handling.
- Demonstrate systematic knowledge of backend and frontend by developing an appropriate application.

LIST OF EXPERIMENTS
1. Writing Java programs by making use of class, interface, package, etc for the following:
   a. Simple programs using control structures
   b. Stack / Queue implementation using Array.
   c. Constructor
   d. Method overloading
   e. Types of inheritance
   f. Overriding and interfaces
   g. Creation of user specific packages
   h. User specific exception handling
2. Write a java program to read the contents of a file and copy to destination file.
3. Write a java multithread program to synchronize the producer consumer problem using Inter Process Communication.
4. Design an applet program to create a color pallet using 5 radio buttons and one Choice box. Provide appropriate event handling to change the background color using radio button and change the foreground color by choice.
5. Design a banner Applet program to scroll the Text “Java programming Lab” using Thread.
6. Design an applet to implement Digital clock using Thread
7. Design an Applet program to create Email registration Form using different awt components. (Minimum 5 components). Write an event handling procedure to validate each field and display appropriate message.
8. Design an applet program to create a Calculator with Grid of Buttons. Write an event handling procedure to activate the buttons with required functionality. Make the contents of text field should not be editable.
9. Design a java GUI Frame to Manipulate the Student details in a database using JDBC. The frame contains the select, insert, delete buttons to do the database activity. The results of database queries are displayed into a Text Area component. The inputs to the queries are collected by using Input Dialog control. Write an event handling procedure to do the above work. Use MS Access to create a student database.
10. Design a Text Editor using java Frame class and the editor must support for creation of new file, open an exiting file and save the contents of Editor area using Menu components. Write an event handling procedure to accomplish the above task.
11. Write a java program to implement the Chat application using Sockets.
12. Write a Java GUI Application to display the content of web page using URL Class.

TOTAL: 45 PERIODS

Softwares Required: Java / Netbeans IDE
MCC37 CASE TOOLS LABORATORY

COURSE OUTCOMES
On successful completion of the course, the students will be able to

- Analyze the functional requirements for a system
- Design, implement and test the programs that make appropriate use of advanced object-oriented facilities.

Mini-Project - I: A Point-of-Sale (POS) System
Develop a computerized application for recording sales and handling payments for a retail store.

Mini-Project - II: Online Bookshop Example
Develop a model like amazon.com or bn.com, design and implement an online bookstore.

Mini-Project - III: A Simulated Company
Simulate a small manufacturing company. The resulting application will enable the user to take out a loan, purchase a machine, and monthly production runs, follow the performance of the company.

Mini-Project - IV: A Multi-Threaded Airport Simulation
Simulate the operations in an airport. Your application should support multiple aircrafts using several runways and gates avoiding collisions/conflicts. Landing: an aircraft uses the runway, lands, and then taxis over to the terminal. Take-Off: an aircraft taxies to the runway and then takes off.

Mini-Project - V: An Automated Community Portal
Develop enterprise intranet portals for sharing information.

Mini-Project - VI: A Content Management System
The goal is to enable non-technical end users to easily publish, access, and share information over the web, while giving administrators and managers complete control over the presentation, style, security, and permissions.

Mini-Project - VII: An Auction Application
Design and implement an auction application that provides auctioning services. It should clearly model the various auctioneers, the bidding process, auctioning etc.

Mini-Project - VIII: A Notes and File Management System
Develop personal notes and documents.

Mini-Project - IX: A Customizable Program Editor
Develop an editor for user interaction.

Mini-Project - X: A Graphics Editor
Design and implement graph editing applications, i.e., applications that include the ability to draw structured and unstructured diagrams.

TOTAL: 45 PERIODS

Softwares Required: Agro UML, Visual paradigm UML, Visual basic, Java
MCC38 GRAPHICS LABORATORY

COURSE OUTCOMES

On successful completion of the course, the students will be able to
- Create simple geometric figures upon a screen.
- Implement transformations of graphical objects in two and three dimensions.
- Employ clipping operations through various algorithms.
- Create image editing applications and animation.
- Apply the use of fractal geometry.

LIST OF EXPERIMENTS

1. Drawing Algorithms
   a. Write a program to draw a line using DDA Algorithm
   b. Write a program to draw a line using Bresenham Algorithm
   c. Write a program to draw a circle using Bresenham Algorithm
   d. Write a program to draw an ellipse using Bresenham Algorithm

2. 2D transformations
   a. Draw a square object. Write a program in C for moving the object diagonally or turn the object clockwise by the user choices.
   b. Draw a circle object. Write a program in C for moving an object vertically or enlarge the object by the user choices.
   c. Draw a triangle object. Write a program in C for moving an object horizontally or turn the object counterclockwise by the user choices.
   d. Draw an oval object. Write a program in C for enlarging an object or spin the object counterclockwise by the user choices.
   e. Draw a polygon object. Write a program in C for shrinking an object or spin the object clockwise by the user choices.
   f. Draw an ellipse object. Write a program in C for resizing or changing the position of the object.

3. 2D composite transformations
   a. Draw a star object. Write a program in C for moving an object diagonally and rotating the object using composite transformation.
   b. Draw a diamond object. Write a program in C for enlarging and rotating the object using composite transformation.

4. Reflection and Shear
   a. Write a program in C for getting the mirror image of an object in X and Y direction
   b. Write a program in C for changing the square to rhombus shape in X and Y direction.

5. Cohen Sutherland line clipping algorithm
   a. Write a line clipping program in C which involves logical operations,
   b. Write a line clipping program in C which has encoding operations.

6. Liang Barsky line clipping algorithm
   a. Write a line clipping program in C which involves clip-test operation.
   b. Write a line clipping program in C which has less multiplications and only one division.

7. Sutherland – Hodgeman Polygon clipping Algorithm
8. 3D transformations
   a. Draw a cylinder object. Write a program in C for moving the object diagonally or turn the object clockwise by the user choices.
   b. Draw a cube object. Write a program in C for moving an object vertically or enlarge the object by the user choices.
   c. Draw a cone object. Write a program in C for moving an object horizontally or shrink the object by the user choices.
   d. Draw a cube object. Write a program in C for rotating the object in X and Y direction.
   e. Draw a hexagon object. Write a program in C for rotating the object in X or Z direction.
   f. Draw a pyramid object. Write a program in C for rotating the object in Y or X direction

9. 3D composite transformations
   a. Draw a cube object. Write a program in C for moving and resizing the object.

10. Create an application for image editing, enhancement, manipulation by using editing tools, layers, filters, special effects and color modes using photoshop.
    a. Create an effect such that one image overlaps another.
    b. Create a selective color change effect.
    c. Type your name in caps. Apply an image into that.
    d. Open a flower. Copy it and paste it four times. Give different colors to each
    e. Open two people’s images. Cut one’s head and paste it into another
    f. Design a greeting card at least using 3 effects
    g. Create a mirror image of an object
    h. Print a watermark effect
    i. Create an effect such that a baby’s face smiling inside a flower
    j. Adjust the features of a person’s face
    k. Open an Image, try to bring the effect of painting
    l. Repaint the damaged images
    m. Make an image glowing
    n. Blur and sharpen an image.
    o. Take a grayscale image. Change the color only in the face

11. Create an animation using Image Ready.
    a. Two people walking from opposite side handshaking.
    b. Changing colors of a rose
    c. Printing our college name one letter by another
    d. Rising a sun slowly from sea
    e. Blinking of an eye of a child.

12. Fractal Images
    a. Generating fractal images with self-similarity
    b. Generating fractal images with iterated functions.

Software Required: C, Photoshop, Image Ready.

TOTAL: 45 PERIODS
MCC41 NETWORK PROGRAMMING L T P C
3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Develop client-server communication using TCP and UDP sockets.
• Develop iterative and concurrent server using TCP.
• Implement I/O multiplexing using select and poll functions.
• Implement concurrent server using threads.
• Develop connection less client-server communication using Java.

UNIT I ELEMENTARY TCP SOCKETS 9

UNIT II APPLICATION DEVELOPMENT 9

UNIT III SOCKET OPTIONS, ELEMENTARY UDP SOCKETS 9

UNIT IV ADVANCED SOCKETS 9

UNIT V NETWORK PROGRAMMING IN JAVA 9

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
MCC42 WEB PROGRAMMING

3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
- Design and implement complete applications over the web using client and server side scripting languages.
- Interconnect clients and server using interconnectivity techniques.
- Retrieve data from a database and present it in a web page.
- Modify, add, and delete data in a database through a web page.

UNIT I INTRODUCTION
Client and Server concepts, Tiered architecture, WWW, IP Address, URL, URI, URN, Domain Name System, Internet protocols and applications: TCP, UDP, FTP, SMTP, POP, ICMP, SNMP; Types of Networks and applications, HTML.

UNIT II DYNAMIC HTML

UNIT III SCRIPTS and APPLETS

UNIT IV SERVLETS

UNIT V ASP and JSP
ASP basics – ASP objects – ASP applications – JSP Programming – JSP objects – Applications – PHP – MySQL.

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
MCC43    COMPILER DESIGN    L T P C
3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Distinguish formal and practical properties of different approaches to parsing.
• Understand and implement various techniques to parse source code.
• Implement a basic compiler.
• Analyze the optimization technique on the intermediate representation.

UNIT I  LEXICAL ANALYSIS  9

UNIT II  SYNTAX ANALYSIS  9

UNIT III  INTERMEDIATE CODE GENERATION  9

UNIT IV  CODE OPTIMIZATION  9

UNIT V  CODE GENERATION  9

TOTAL: 45 PERIODS

TEXT BOOK

REFERENCES
MCC44 \hspace{1cm} NETWORK PROGRAMMING LABORATORY \hspace{1cm} L\ T\ P\ C
\hspace{4cm} 0\ 0\ 3\ 2

**COURSE OUTCOMES**
On successful completion of the course, the students will be able to
- Analyze and develop client/server communication using connection oriented and connection less protocols.
- Demonstrate domain name system.
- Competent with application development and debugging in Unix environments.
- Demonstrate remote method invocation and remote procedure call.

**LIST OF EXPERIMENTS**

1. Implement a client/server user-level application using TCP sockets in C. The Server application has to accept string from client and echo the received string. For example, when client sends “Welcome to Network programming Lab”, Server replies with “Welcome to Network programming Lab.”

2. Implement a client/server user-level application using TCP sockets in C. The Server application has to support ‘n’ number of clients, but one by one. The Server application has to accept string from client and replies with received string. For example, when client sends “Iterative Server”, Server replies with “Iterative server”.

3. Implement a client/server user-level application using TCP sockets in C. The Server application has to support ‘n’ number of clients simultaneously. The Server application has to accept string from client and replies with received string. For example, when client sends “concurrent sever”, Server replies with “concurrent server”.

4. Implement a client/server user-level application using TCP sockets in C. The Server application must be able to chat with ‘n’ number of clients simultaneously.

5. Implement a client/server user-level application using UDP sockets in C. The Server application must be able to chat with ‘n’ number of clients simultaneously.

6. Implement a client/server user-level application using UDP sockets in C. The Server application has to accept string from client and echo the received string. For example, when client sends “Welcome to Network programming Lab”, Server replies with “Welcome to Network programming Lab.”

7. Implement a client/server user-level application using UDP sockets in C. The Server application has to support ‘n’ number of clients, but one by one. The Server application has to accept string from client and replies with received string. For example, when client sends “Iterative Sever”, Server replies with “Iterative server”.

8. Implement a client/server user-level application using UDP sockets in C. The Server application has to support ‘n’ number of clients simultaneously. The Server application has to accept string from client and replies with received string. For example, when client sends “concurrent sever”, Server replies with “concurrent server”.

9. Write a client/server program wherein the client sends the name of a command to be executed at the server. The server then sends the result back to the client. Implement the above using connection less service.
10. Write a client/server program wherein the client the name of a program to be executed at the server. The server then sends the result back to client. Implement the above using connection oriented service and concurrent server.

11. Write a client server program wherein the client when connects to the server, the server then sends the system data and time to the client. Implement the above using RMI.

12. Develop a client / server communication program in Java using connection less protocol.

13. Write a client / server program wherein the client sends the IP – address and the server responds by sending the corresponding host's name. Use connection less service to implement the above.

14. Write a client / server program wherein the client sends the name of a file and the server returns the contents of the file. Use connection oriented service to implement the above. Implement the server as an iterative server.

15. Write a client / server program wherein the client sends two numbers, the server finds its gcd and returns to the client. Use RPC to implement above.

16. Write client / server program wherein the client sends a Unix Command (like 'ls') and the server returns the result of execution of the command. Use RPC to implement the above.

17. Write a program to implement the following interaction between client and server. User supplies the filename to the client program. Client program in turn sends it to the server; the server reads the contents of a text file and then sends the contents to the client, then the client displays it on the screen. Implement with concurrent TCP server.

**Software Required:** Unix, C and Java

**TOTAL: 45 PERIODS**
MCC45       WEB PROGRAMMING LABORATORY       L T P C
                                      0 0 3 2

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Design Web Pages using Client Side Scripting and DHTML.
• Implement server side languages like Servlets, JSP and ASP.
• Develop web services and E-business applications.

LIST OF EXPERIMENTS
1. Design of image mapping for an image using HTML. Link the documents using an image and
mark the hot spots in the image.
2. Development of web page using cascading Style Sheets (CSS) and implement its types.
3. Create a registration form and do the validation using HTML and Java Script.
4. Design of color palette using java and change the background and foreground of web pages.
5. Implement the types of layouts for different application and develop a simple calculator using
grid layout.
6. Design and develop an application using servlets and interconnect client and server.
7. Invoke HTML form using servlets and create interactivity.
8. Develop a real time application using applets.
9. Implement an Employee payroll processing application using ASP and connect client and server.
10. Develop an Online examination using JSP and display the marks.
11. Implement a web system using JDBC and interconnect client and server using servlets.
12. Design an online shopping web page using JSP and design an interactive online shopping web
site.

TOTAL: 45 PERIODS

Software Required: Java, XML, HTML, Scripting languages, ASP, JSP, Servlets.
MCC46  COMPILER DESIGN LABORATORY  L T P C
0 0 3 2

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Understand the language translation peculiarities by designing complete translator for mini
language.
• Understand the design aspect of various phases of compiler.

LIST OF EXPERIMENTS
1. Implement a lexical analyzer in “C”.
2. Use LEX tool to implement a lexical analyzer.
3. Write a C Program to convert a grammar into finite automata
4. Write a C Program to convert a Regular expression into finite automata
5. Implement a recursive descent parser for an expression grammar that generates
   arithmetic expressions with digits, + and *.
6. Implement LL (1) parser using C Program.
7. Write semantic rules to the YACC program in problem 5 and implement a calculator that takes
   an expression with digits, + and * and prints its value.
8. Implement the front end of a compiler that generates the three address code for a simple
   language with: one data type integer, arithmetic operators, relational operators, variable
   declaration statement, one conditional construct, one iterative construct and assignment
   statement.
9. Implement the back end of the compiler which takes the three address code generated in
   problems 7 and 8, and produces the 8086 assembly language instructions that can be assembled
   and run using a 8086 assembler. The target assembly instructions can be simple move, add, sub,
   and jump.
10. Implement code optimization technique to improve the quality of the program.

TOTAL: 45 PERIODS

Software Required: TURBO C, LEX, YACC
MCE4A  NUMERICAL AND STATISTICAL METHODS  L T P C
3 0 0 3

COURSE OUTCOMES
At the end of this course, students will be able to
• Use numerical techniques for solving linear system of equations.
• Understand and utilize the problems in numerical differentiation and numerical integration.
• Demonstrate the utility of numerical techniques of ordinary differential equations.
• Apply the concepts of estimation (confidence intervals) and hypothesis testing for population averages and percentages.
• Analyze the appropriate tabular for displaying design of experiments.

UNIT I  SOLUTION OF EQUATIONS AND EIGENVALUE PROBLEMS  9

UNIT II  INTERPOLATION, NUMERICAL DIFFERENTIATION AND NUMERICAL INTEGRATION  9
Lagrange’s and Newton’s divided difference interpolation – Newton’s forward and backward difference interpolation – Approximation of derivatives using interpolation polynomials – Numerical integration using Trapezoidal and Simpson’s 1/3 rules for single and double integration.

UNIT III  NUMERICAL SOLUTION OF ORDINARY DIFFERENTIAL EQUATIONS  9

UNIT IV  TESTING OF HYPOTHESIS  9
Sampling distributions – Tests for single mean, Proportion, Difference of means (large and small samples) – Tests for single variance and equality of variances – Chi-square test for goodness of fit – Independence of attributes.

UNIT V  DESIGN OF EXPERIMENTS  9
Completely randomized design – Randomized block design – Latin square design.

TOTAL: 45 PERIODS

TEXTBOOKS

REFERENCES
MCE4B ELECTRONIC COMMERCE

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Explore basic Internet Transactions features
• Apply the Security Technologies in Electronic Commerce
• Acquire knowledge about the strategies to develop Electronic Commerce websites and Payment Systems
• Acquire knowledge about the environment of Electronic commerce, Techniques and Tools of Electronic Data Interchange

UNIT I INTRODUCTION 6

UNIT II SECURITY TECHNOLOGIES 9

UNIT III ELECTRONIC PAYMENT METHODS 9

UNIT IV ELECTRONIC COMMERCE PROVIDERS 9

UNIT V ONLINE COMMERCE ENVIRONMENTS 12

TOTAL: 45 PERIODS

TEXT BOOK

REFERENCES
MCE4C    INFORMATION SYSTEMS    L T P C
          3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Identify and analyze Information System Management and requirements in Decision Making
• Communicate to both business and IT professionals
• Implement practical applications of Information Systems in Business and Society

UNIT I    INFORMATION SYSTEM AND ORGANIZATION    9

UNIT II    INFORMATION TECHNOLOGY INFRASTRUCTURE    9

UNIT III    INFORMATION SYSTEMS AND DECISION MAKING    9

UNIT IV    SYSTEM DEVELOPMENT APPROACHES    9

UNIT V    INFORMATION SYSTEM DEVELOPMENT TOOLS AND APPLICATIONS    9
Computer-Aided System Engineering (CASE) Tools – Business – Accounting – Health – Academic and Social Services

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
MCE4D WEB GRAPHICS L T P C
3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Design the web sites and create images and animated pictures.
• Design logo, brochure and images.
• Develop animations, graphics and movies using Windows Media Player.

UNIT I INTRODUCTION
Basic web graphics – Web designing, Image maps – Pixels, Resolutions, Principles of Graphic design,
Raster Graphics Vs Vector Graphics, Introduction to CorelDrar – Basic Drawing Skills – Using
Text – Working with Objects – Adding Special Effects – Creating Output.

UNIT II RASTER IMAGE EDITING SOFTWARE
Creating new Images – Brushes – Grids and Guides – Gradients – Scaling Images – Moving and
Merging Layers – Tool Palette – Dialogs – Masking – Filters – Adding text to images – Designing
icons and background images.

UNIT III VECTOR IMAGE HANDLING
Tweening – Motion guide – Masking – Frame by Frame animation – Onion Skin Effect – Creating
special effects – Text effects and animation – Action scripts.

UNIT IV MULTIMEDIA
Creating clippings – Animations with sound effects – Adding audio or Video – Windows Media
Player ActiveX Control – Agent control – Embedding VRML in a web page – Real Player ActiveX
control.

UNIT V APPLICATIONS
Development of interactive web applications with a particular theme using vector graphics and raster
graphics concepts – Animations and Interaction.

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
1. Adobe creative team, “Adobe Photoshop Elements 7 and Adobe Premiere Elements
MCE4E ADVANCED DATABASES  L T P C 3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to

- Explain and evaluate the fundamental theories and requirements that influence the design of modern database systems.
- Assess and apply database functions and packages suitable for distributed database development.
- Critically evaluate alternative designs and architectures for databases and data warehouses.
- Analyze the background processes involved in queries and transactions, and explain how these impacts on spatial and temporal database operation and design.

UNIT I DATA WAREHOUSING  8

UNIT II DISTRIBUTED DATABASES  9

UNIT III OBJECT ORIENTED DATABASES  10

UNIT IV SPATIAL AND TEMPORAL DATABASES  9

UNIT V RESEARCH TRENDS  9

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
MCE4F   SOFTWARE QUALITY MANAGEMENT   LT P C
         3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Use quality models to identify and specify the quality attributes a software system must satisfy
• Find the interrelation between product quality and process quality
• Apply the product and process quality control techniques and able to create a quality product with standards

UNIT I   FUNDAMENTALS OF SOFTWARE QUALITY ENGINEERING   9

UNIT II   DEVELOPMENTS IN MEASURING QUALITY   9

UNIT III   QUALITY MANAGEMENT SYSTEM   9

UNIT IV   PRINCIPLES AND PRACTICES IN QMS   9

UNIT V   MEASURES AND METRICS IN PROCESS AND PROJECT DOMAINS   9

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES
MCE4G                  TCP / IP DESIGN AND IMPLEMENTATION                        L   T   P   C  
3  0  0  3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Develop the practical experience of IP addresses, and the fundamentals of IP routing
• Build the trade-offs between UDP and TCP and its uses
• Implement the details of IP and TCP operations

UNIT I      INTRODUCTION                             9
Internetworking concepts and architectural model – Classful Internet address – CIDR – Subnetting and

UNIT II      TCP                                                                                                                             9
Services – Header – Connection establishment and termination – Interactive data flow – Bulk data
flow – Timeout and retransmission – Persist timer – Keepalive timer – Futures and performance –
TCP Undiagnosed Problems.

UNIT III      IP IMPLEMENTATION                                                                  9
IP global software organization – Routing table – Routing algorithms – Fragmentation and reassembly
– Error processing (ICMP) – Multicast Processing (IGMP) Using IP Multicasts with Windows
Sockets Programs – Configuration Parameters Using DHCP.

UNIT IV      TCP IMPLEMENTATION I                                                                                   9
Data structure and input processing – Transmission control blocks – Segment format – Comparison
– Finite state machine implementation – Output processing – Mutual exclusion – Computing the TCP
data length – Integration of Windows NT DNS and WINS Servers.

UNIT V       TCP IMPLEMENTATION II                                                                                     9
Timers – Events and messages – Timer process – Deleting and inserting timer event – Flow control
and adaptive retransmission – Congestion avoidance and control – Urgent data processing and push
function – Architecture of Microsoft TCP/IP for Windows NT, Nagle Algorithm.

TOTAL: 45 PERIODS

TEXT BOOK

REFERENCES
MCE4H    DISTRIBUTED SYSTEMS                      LT P C
3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Articulate relative advantages and drawbacks of the main types of formal models for a
distributed system
• Describe algorithms and/or impossibility results for typical abstract problems in distributed
computing and Operating system
• Identify appropriate complexity measures and analyze solutions to Fault tolerance in
distributed environment

UNIT I  COMMUNICATION IN DISTRIBUTED ENVIRONMENT                      9
Introduction – Various Paradigms in Distributed Applications – Remote Procedure Call – Remote
Object Invocation – Message Oriented Communication – Unicasting Multicasting and Broadcasting –
Client-server communication – Group Communication – Interprocess Communication.

UNIT II DISTRIBUTED OPERATING SYSTEMS                      11
Issues in Distributed Operating System – Threads in Distributed Systems – Clock Synchronization –
Transactions – Distributed Deadlock – Agreement Protocols .

UNIT III DISTRIBUTED RESOURCE MANAGEMENT                      11
Distributed Shared Memory – Data-Centric Consistency Models – Client-Centric Consistency Models

UNIT IV FAULT TOLERANCE AND CONSENSUS                      7
Introduction to Fault Tolerance – Distributed Commit Protocols – Byzantine Fault Tolerance –
Impossibilities in Fault Tolerance – Distributed debugging.

UNIT V CASE STUDIES                      7
Ivy – Munin – CORBA – COM+ – Distributed Coordination – Based System – JINI.

TOTAL: 45 PERIODS

TEXT BOOK
1. George Coulouris, Jean Dollimore, Tim Kindberg, “Distributed Systems Concepts and

REFERENCES
1. Hagit Attiya and Jennifer Welch, “Distributed Computing: Fundamentals, Simulations and
3. M.L.Liu, “Distributed Computing Principles and Applications”, Pearson / Addison Wesley,
   2004.
MCE4J  UNIX INTERNALS  L T P C  3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
- Work with UNIX shell and file system.
- Develop skills to experience common UNIX programming tools to create UNIX applications.
- Work with UNIX files, processes, signals, sockets, and various other SystemV constructs.
- Implement memory allocation in response to specific requests, to manage and reclaim memory.

UNIT I  UNIX SYSTEM STRUCTURES  9

UNIT II  INTERNAL REPRESENTATION OF FILES  9
Inode – Structure of a regular file – Directories – Conversion of a path names to an Inode – Superblock – Inode assignment to a file – Allocation of disk blocks – Other file types.

UNIT III  SYSTEM CALLS FOR THE FILE SYSTEM  9

UNIT IV  PROCESSES  9

UNIT V  MEMORY MANAGEMENT  9

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCE
MCE4K VISUAL PROGRAMMING LT P C 3 0 0 3

COURSE OUTCOMES
On successful completion of the course, the students will be able to
• Develop Windows Application Programming Interface (API).
• Build well structured GUI programming using Microsoft Foundation Classes.
• Develop simple applications using Visual C++.
• Integrate the media content with the programming environment.

UNIT I WINDOWS PROGRAMMING 10

UNIT II VISUAL C++ PROGRAMMING FUNDAMENTALS 9

UNIT III DOCUMENT VIEW ARCHITECTURE 9

UNIT IV CONTROLS 9

UNIT V ADVANCED CONCEPTS 8

TOTAL: 45 PERIODS

TEXT BOOKS

REFERENCES