VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM

Master of Computer Applications (MCA)

Scheme and Syllabus
(With effect from 2013-2014)
## SCHEME OF TEACHING AND EXAMINATION
### MASTER OF COMPUTER APPLICATIONS

### I SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination</th>
<th>IA</th>
<th>Exam</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory Practical</td>
<td>Duration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>13MCA11</td>
<td>Problem Solving using C</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>13MCA12</td>
<td>Discrete Mathematics Structure</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>13MCA13</td>
<td>Fundamentals of Computer Organization</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>13MCA14</td>
<td>Introduction to Unix</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>13MCA15</td>
<td>Introduction to Web Technologies</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>13MCA16</td>
<td>C Programming Laboratory</td>
<td>- 03 03</td>
<td>50 50 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>13MCA17</td>
<td>Unix Programming Laboratory</td>
<td>- 03 03</td>
<td>50 50 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>13MCA18</td>
<td>Web Programming Laboratory</td>
<td>- 03 03</td>
<td>50 50 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total</td>
<td>20</td>
<td>09</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>650</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1050</td>
</tr>
</tbody>
</table>

### II SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Week</th>
<th>Examination</th>
<th>IA</th>
<th>Exam</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory Practical</td>
<td>Duration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>13MCA21</td>
<td>Data Structures Using C</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>13MCA22</td>
<td>Object Oriented Programming using C++</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>13MCA23</td>
<td>Operating Systems</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>13MCA24</td>
<td>System Programming</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>13MCA25</td>
<td>Database Management Systems</td>
<td>04 - 03</td>
<td>50 100 150</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>13MCA26</td>
<td>Data Structures Using C Lab</td>
<td>- 03 03</td>
<td>50 50 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>13MCA27</td>
<td>Database Laboratory</td>
<td>- 03 03</td>
<td>50 50 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>13MCA28</td>
<td>OOP Using C++ Laboratory</td>
<td>- 03 03</td>
<td>50 50 100</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Total</td>
<td>20</td>
<td>09</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>650</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1050</td>
</tr>
</tbody>
</table>
# SCHEME OF TEACHING AND EXAMINATION
## MASTER OF COMPUTER APPLICATIONS

### III SEMESTER

<table>
<thead>
<tr>
<th>S. No</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Duration</th>
<th>Examinations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory / Practical</td>
<td>IA / Exam / Total</td>
</tr>
<tr>
<td>1</td>
<td>13MCA31</td>
<td>Computer Networks</td>
<td>04 / 03</td>
<td>50 / 100 / 150</td>
</tr>
<tr>
<td>2</td>
<td>13MCA32</td>
<td>Programming using Java</td>
<td>04 / 03</td>
<td>50 / 100 / 150</td>
</tr>
<tr>
<td>3</td>
<td>13MCA33</td>
<td>Software Engineering</td>
<td>04 / 03</td>
<td>50 / 100 / 150</td>
</tr>
<tr>
<td>4</td>
<td>13MCA34</td>
<td>Computer Graphics</td>
<td>04 / 03</td>
<td>50 / 100 / 150</td>
</tr>
<tr>
<td>5</td>
<td>13MCA35</td>
<td>Elective-I</td>
<td>04 / 03</td>
<td>50 / 100 / 150</td>
</tr>
<tr>
<td>6</td>
<td>13MCA36</td>
<td>Java Programming Lab</td>
<td>- / 03</td>
<td>50 / 100 / 100</td>
</tr>
<tr>
<td>7</td>
<td>13MCA37</td>
<td>Computer Graphics Lab</td>
<td>- / 03</td>
<td>50 / 100 / 100</td>
</tr>
<tr>
<td>8</td>
<td>13MCA38</td>
<td>Computer Networks Lab</td>
<td>- / 03</td>
<td>50 / 100 / 100</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td>20 / 09</td>
<td>40 / 650 / 1050</td>
</tr>
</tbody>
</table>

**Elective-1**

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA351</td>
<td>UNIX system Programming</td>
</tr>
<tr>
<td>13MCA352</td>
<td>Advanced Topics in DBMS</td>
</tr>
<tr>
<td>13MCA353</td>
<td>Basics of MIS &amp; E-Commerce</td>
</tr>
<tr>
<td>13MCA354</td>
<td>Operations Research</td>
</tr>
<tr>
<td>13MCA355</td>
<td>Principles of User Interface Design</td>
</tr>
<tr>
<td>13MCA356</td>
<td>Probability Statistics &amp; Numerical Methods</td>
</tr>
</tbody>
</table>
## SCHEME OF TEACHING AND EXAMINATION
### MASTER OF COMPUTER APPLICATIONS

### IV SEMESTER

<table>
<thead>
<tr>
<th>S. No</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Duration</th>
<th>Examinati Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>IA</td>
<td>Exam</td>
</tr>
<tr>
<td>1</td>
<td>13MCA41</td>
<td>Analysis &amp; Design of Algorithms</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>13MCA42</td>
<td>Advanced Java Programming</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>13MCA43</td>
<td>Advanced Web Programming</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>13MCA44</td>
<td>Elective-2</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>13MCA45</td>
<td>Elective-3</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>13MCA46</td>
<td>Algorithms Lab</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>7</td>
<td>13MCA47</td>
<td>Advanced Java Lab</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>8</td>
<td>13MCA48</td>
<td>Mini Project-I</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>TOTAL</strong></td>
<td><strong>20</strong></td>
<td><strong>09</strong></td>
</tr>
</tbody>
</table>

### Elective-2

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA441</td>
<td>Advanced Computer Networks</td>
</tr>
<tr>
<td>13MCA442</td>
<td>Data Warehousing &amp; Data Mining</td>
</tr>
<tr>
<td>13MCA443</td>
<td>Mobile Computing and Wireless Communications</td>
</tr>
<tr>
<td>13MCA444</td>
<td>Software Testing and Practices</td>
</tr>
<tr>
<td>13MCA445</td>
<td>Theory of Computation (FAFL)</td>
</tr>
<tr>
<td>13MCA446</td>
<td>Digital Image Processing</td>
</tr>
</tbody>
</table>

### Elective 3

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA451</td>
<td>Cryptography &amp; Network Security</td>
</tr>
<tr>
<td>13MCA452</td>
<td>Network Management</td>
</tr>
<tr>
<td>13MCA453</td>
<td>NOSQL</td>
</tr>
<tr>
<td>13MCA454</td>
<td>Software Architectures</td>
</tr>
<tr>
<td>13MCA455</td>
<td>Enterprise Resource Planning (ERP)</td>
</tr>
<tr>
<td>13MCA456</td>
<td>Mobile Applications</td>
</tr>
</tbody>
</table>
## SCHEME OF TEACHING AND EXAMINATION
### MASTER OF COMPUTER APPLICATIONS

#### V SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs / Duration</th>
<th>Examinatio Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>IA</td>
<td>Exam</td>
</tr>
<tr>
<td>1</td>
<td>13MCA51</td>
<td>Object-Oriented Modeling and Design Patterns</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>13MCA52</td>
<td>System Simulation and Modeling</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>13MCA54</td>
<td>Elective-4</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>13MCA55</td>
<td>Elective-5</td>
<td>04</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>13MCA56</td>
<td>Software Design Laboratory</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>7</td>
<td>13MCA57</td>
<td>.Net Laboratory</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td>8</td>
<td>13MCA58</td>
<td>Mini Project -II</td>
<td>-</td>
<td>03</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td><strong>400</strong></td>
</tr>
</tbody>
</table>

#### Elective-4

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA541</td>
<td>Mobile and Adhoc Sensor Networks</td>
</tr>
<tr>
<td>13MCA542</td>
<td>Parallel Computing</td>
</tr>
<tr>
<td>13MCA543</td>
<td>Multimedia systems</td>
</tr>
<tr>
<td>13MCA544</td>
<td>Pattern Recognition</td>
</tr>
<tr>
<td>13MCA545</td>
<td>Services Oriented Architecture</td>
</tr>
<tr>
<td>13MCA546</td>
<td>Compiler Design</td>
</tr>
</tbody>
</table>

#### Elective-5

<table>
<thead>
<tr>
<th>Subject Code</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA551</td>
<td>Cloud Computing</td>
</tr>
<tr>
<td>13MCA552</td>
<td>Web2.0 and Rich Internet Applications</td>
</tr>
<tr>
<td>13MCA553</td>
<td>Information Retrieval &amp; Search Engines</td>
</tr>
<tr>
<td>13MCA554</td>
<td>Soft Computing</td>
</tr>
<tr>
<td>13MCA555</td>
<td>Storage Area Network</td>
</tr>
<tr>
<td>13MCA556</td>
<td>Software Project Management</td>
</tr>
</tbody>
</table>
# SCHEME OF TEACHING AND EXAMINATION
## MASTER OF COMPUTER APPLICATIONS
### VI SEMESTER

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Subject Code</th>
<th>Title</th>
<th>Teaching Hrs/Week</th>
<th>Examination</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Theory</td>
<td>Practical</td>
</tr>
<tr>
<td>1</td>
<td>13MCA61</td>
<td>Project</td>
<td>-</td>
<td>03</td>
</tr>
</tbody>
</table>

**GENERAL NOTE:**

1. Students have to register for one Elective subject from each of the Five Elective Groups.
   (One elective from III Semester, two electives from IV Semester, two electives from V Semester)

2. Change of program during lab examinations is not permitted because problems are given from the predefined list only.
SEMESTER I

Problem Solving Using C

Sub. Code : 13MCA11 IA Marks : 50
Hrs/Week : 4 Exam Hours : 03
Total Hours : 52 Exam Marks : 100

Unit 1 9 Hours
Algorithms, Flow Charts, C structure, Variables, Data types, Constants, Declarations, Operators, Precedence, Associativity, Order of evaluation, Type conversion, Storage classes, Programming Examples

Unit 2 10 Hours
Input and output statements – scanf, getchar, gets, printf, putchar, puts; Control Statements – if, else-if, switch, Control Structures – while, for, do-while, break and continue, goto, Programming Examples

Unit 3 8 Hours
Arrays – Single dimension, Two dimensional, Multi dimensional Arrays, Strings, Programming Examples

Unit 4 10 Hours
Functions, Categories of functions, Pointers, Pointer arithmetic, Call by value, Pointer Expression, Pointer as function arguments, recursion, Passing arrays to functions, passing strings to functions, Call by reference, Functions returning pointers, Pointers to functions, Programming Examples

Unit 5 5 Hours
Structures and Unions – defining, declaring, initialization, accessing, comparing, operations on individual members; array of structures, structures within structures, structures and functions, pointers and structures, bit fields, Programming Examples

Unit 6 10 Hours
Files – defining, opening, closing, input and output operations, error handling, random access; Command line arguments; Dynamic Memory Allocation – definition, malloc, calloc, realloc, free, dynamic arrays; Preprocessor – definition, macro substitution, file inclusion, compiler control directives, Programming Examples

Text Books
1. Let us C, Yashwant Kanetkar, BPB Publications
2. Programming with C, Balaguruswamy
3. The C Programming Language, Brian W Kernighan, Dennis M Ritchie, PHI, 2nd Edition

Reference Books
2. Simplifying C, Harshal Arolkar, Sonal Jain, Wiley Publications
3. Head First C, David Griffiths, & Dawn Griffiths, O’Riley.
Discrete Mathematical Structures

Subject Code : 13MCA12                                      I.A. Marks : 50
Hours/Week : 04                                          Exam Hours : 03
Total Hours : 52                                      Exam Marks : 100

Fundamentals of Logic  14 Hours
Basic Connectives and Truth Tables, Logic Equivalence :The laws of Logic,
Logical Implications: Rules of Inference, The use of Quantifiers, Quantifier Definitions,
Proofs of Theorems.

Set Theory 7 Hours
Sets and Subsets, Set Operations and the Laws of Set Theory, Counting and Venn
Diagrams, Principles of Inclusion and Exclusion, The rules of sum and product,
Permutations and Combinations with repetition

Properties of Integers and Recurrence 7 Hours
Mathematical Induction, Recursive definitions, The Greatest Common Divisor
Euclidian Algorithms, The first order Linear recurrence relation.

Relations and Functions 14 Hours
Cartesian products and Relations, Functions-Plain and One-to-One, Onto Functions,
Stirling Numbers and the Second Kind, Special functions, The Pigeon-hole principle,
Function composition and inverse functions. Properties of Relations, Computer recognition-
Zero One Matrices and Directed graphs, Posets and Hasse Diagrams, Equivalence relation
and Partitions, lattices.

Graph Theory and Trees 10 Hours
Terminology, Definitions, Properties and Examples, Connectivity and Adjacency, Euler
and Hamilton, Representation and Isomorphism, Planarity and Chromatic Number, Directed
Graphs and Weighted Graphs, Rooted Trees, Trees and Sorting

Text Books
   Introduction” 5th Edition, Pearson Education, 2004 (Chapter1: 1.1-1.4, Chapter 2: 2.1-
   2.5, Chapter 3: 3.1-3.4, Chapter 4: 4.1-4.2,4.4, Chapter 5:5.1-5.6, Chapter 7:7.1-,
   7.4,7.6, Chapter 10:10.1, Chapter 12: 12.1-12.4)
2. Eric Gosset “Discrete Mathematics with Proof” Wiley India, 2nd Edition,(Chapter
   10:10.1-10.6)

Reference books
   Learning, 2004
Fundamentals Of Computer Organization

Subject Code: 13MCA13       I.A. Marks : 50
Hours/Week : 04         Exam Hours : 03
Total Hours : 52         Exam Marks : 100

Binary Systems
6 Hours

Combinational Logic and Arithmetic Circuits
14 Hours

Sequential Logic
6 Hours
Introduction, different types of Flip – Flops, Triggering of Flip- Flops, Registers, Shift Registers, Ripple counter and Synchronous Counter.

Basic Structure of Computers
6 hours

Machine Instruction and Programs
7 Hours
Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes, Examples from Assembly Language Programming.

Input/Output Organization
7 Hours
Accessing I/O Devices, Interrupts, DMA, Processor Example, Buses. Case study of IA32 Intel processor

The Memory System
6 Hours

Text Books:

Reference Books:
Introduction to UNIX

Sub Code : 13MCA14
Hrs/Week : 04
Total Hours : 52
IA Marks : 50
Exam Hours : 03
Exam Marks : 100

Introduction of UNIX 

Introduction to the Shell 
Introduction to Shell Scripting, Shell Scripts, read, Command Line Arguments, Exit Status of a Command, The Logical Operators && and ||, exit, if, and case conditions, expr, sleep and wait, while, until, for, $, @, redirection. The here document, set, trap, Sample Validation and Data Entry Scripts.

Basic File Attributes 
ls – l, the –d option, File Permissions, chmod, Security and File Permission, users and groups, security level, changing permission, user masks, changing ownership and group, File Attributes, More file attributes: hard link, symbolic link, umask, find

Simple Filters 
Pr, head, tail, cut, paste, sort, uniq, tr commands, Filters using Regular Expression : grep & sed grep, Regular Expression, egrep, fgrep, sed instruction, Line Addressing, Inserting and Changing Text, Context addressing, writing selected lines to a file, the –f option, Substitution, Prosperities of Regular Expressions Context addressing, writing selected lines to a file, the –f option, Substitution, Properties of Regular Expressions

Awk-Advanced Filters 
Simple awk Filtering, Splitting a Line into Fields, printf, the Logical and Relational Operators, Number Processing, Variables, The –f option, BEGIN and END positional Parameters, get line, Built-in variables, Arrays, Functions, Interface with the Shell, Control Flow, Advanced Shell Programming, The sh command, export, cd, the Command, expr, Conditional Parameter Substitution, Merging Streams, Shell Functions, eval, Exec Statement

The Process 
Process basics, PS, internal and external commands, running jobs in background, nice, at and batch, cron, time commands, Essential System Administration root, administrator’s privileges, startup & shutdown, managing disk space, cpio, tar, Customizing the Environment : System Variables, profile, sty, PWD, Aliases, Command History, On-line Command Editing

Advanced System Administration 
Case Study: emacs editor and any one distribution of Linux

Text Book: 
1. Your UNIX-The Ultimate Guide, Sumitabha Das, Tata McGraw Hill,

Reference Book: 
1. “Unix Shell Programming”, Yashwant Kanetkar,
Introduction to Web Technologies

Subject Code: 13MCA15
Hours/Week: 4
Total Hours: 52
I.A. Marks : 50
Exam Marks : 100
Exam Hours : 3

Fundamentals
Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, the Web Programmers Toolbox.

Web Foundations

Introduction to XHTML

The Basics of JavaScript:
Overview of JavaScript, Object orientation and JavaScript, Syntactic characteristics, Primitives, operations, and expressions, Screen output and keyboard input, Control statements, Object creation and modification, Arrays, Functions, Constructors, Pattern matching using regular expressions, Errors in scripts, Examples.

JavaScript and HTML Documents

Dynamic Documents with JavaScript:
Introduction, Positioning Elements, Moving Elements, Element Visibility, Changing Colors and Fonts, Dynamic Content, Stacking Elements, Locating the Mouse Cursor, Reacting to a Mouse Click, Slow Movement of Elements, Dragging and Dropping Elements.

Introduction to XML

The Basics of Perl
Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching, File input and output; Examples. Using Perl for CGI Programming: The Common Gateway Interface; CGI linkage; Query string format; CGI.pm module; A survey example; Cookies.
Text Books:

Reference Books:
1. a. Convert degrees into Fahrenheit and vice versa  
b. Calculate the salary of an employee given his basic pay, HRA = 10% of basic pay, TA=5% of his basic pay and deductions IT = 2.5% of his basic pay

2. a. Check whether the given number is perfect number  
   Defn: A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself.  
   Example - The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6.  
   b. Solve quadratic equations for the given values of a,b,c.

3. a. Generate all Amstrong numbers upto n.  
   Defn: An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself.  
   Example - 371 is an Armstrong number, since 3**3 + 7**3 + 1**3 = 371.  
   b. Convert a decimal number to a hexadecimal number.

4. Write a menu driven C program to  
   a. Insert an element into an array  
   b. Delete an element from the array (first occurrence)

5. Write a Menu Driven C Program to  
   a. Accept a string from the user  
   b. Encode the string.  
   c. Decode the string  
   Apply the following procedure to encode it.  
   1. Convert each character in a string to its ASCII value.  
   2. Add an integer value to it and display the encoded string  
   3. Decode the string using reverse procedure and display.

6. Write a C program to multiply two matrices that satisfy the constraint of matrix multiplication

7. Write a C program to find the saddle point of a matrix.  
   Defn: Given a RxC Matrix, A, i.e. R rows and C columns we define a Saddle-Point as Saddle_Pt (A) for a row I and column j is that A(i,j) that is the minimum of Row i and the maximum of Col j.

8. Write a C program to implement a magic square of size n.  
   Defn: A magic square is an arrangement of numbers (usually integers) in a square grid, where the numbers in each row, and in each column, and the numbers that run diagonally in both directions, all add up to the same number.
9. Write a Menu driven C program to
   a. Accept two numbers n and m
   b. Sum of all integers ranging from n to m
   c. Sum of all odd integers ranging from n to m
   d. Sum of all even integers ranging from n to m
   Display an error message if n > m. Create functions for each of the options.

10. Write a Menu Driven C Program to implement the following using recursion
    a. Factorial of a number
    b. Fibonacci series

11. Create a structure Complex Number having real and imaginary part as properties. Write functions to add and subtract the two complex numbers.

12. Define a structure called student having the properties of student_id, student name and branch of the student with a sub structure of marks of 3 subjects. Write a Menu Driven C Program to
    a. Add new student detail
    b. Delete a student detail
    c. Display all student details
    d. Display the name of the student with the best mark
    e. Display the name of the student with the worst mark
    f. Display the average marks scored by the students

13. a. Write a C Program to remove all white spaces and newline characters from a file.
    b. Find whether a given word exists in the file. If it exists display the location of the word

14. Write a C program to copy one file content to another file without using inbuilt functions.

Note: In the examination each student should do one question out of the above 14 questions
A. Explore the unix environment.

B. Explore vi editor with vim tutor. Perform the following operations using vi editor, but not limited to:

1. insert character, delete character, replace
2. Save the file and continue working
3. save the file a exit the editor
4. quit the editor
5. quit without saving the file
6. rename a file
7. insert lines, delete lines,
8. set line numbers
9. search for a pattern
10. move forward and backward

1a. Write a shell script that takes a valid directory name as an argument and recursively descend all the sub-directories, finds the maximum length of any file in that hierarchy and writes this maximum value to the standard output.

b. Write a shell script that accepts a path name and creates all the components in that path name as directories. For example, if the script is named mpc, then the command mpc a/b/c/d should create directories a, a/b, a/b/c, a/b/c/d.

2a. Write a shell script that accepts two file names as arguments, checks if the permissions for these files are identical and if the permissions are identical, output common permissions and otherwise output each file name followed by its permissions.

b. Write a shell script which accepts valid log-in names as arguments and prints their corresponding home directories, if no arguments are specified, print a suitable error message.

3a. Create a script file called file-properties that reads a file name entered and outputs its properties.

b. Write shell script to implement terminal locking (similar to the lock command). It should prompt the user for a password. After accepting the password entered by the user, it must prompt again for the matching password as confirmation and if match occurs, it must lock the keyword until a matching password is entered again by the user. Note that the script must be written to disregard BREAK, control-D. No time limit need be implemented for the lock duration.

4a. Write a shell script that accept one or more filenames as argument and convert all of them to uppercase, provided they exist in current directory.

b. Write a shell script that displays all the links to a file specified as the first argument to the script. The second argument, which is optional, can be used to specify in which the search is to begin. If this second argument is not present, the search is to begin in current working directory. In either case, the starting directory as well as all its subdirectories at
all levels must be searched. The script need not include any error checking.

5a. Write a shell script that accepts as filename as argument and display its creation time if file exist and if it does not send output error message.
b. Write a shell script to display the calendar for current month with current date replaced by * or ** depending on whether the date has one digit or two digits.

6a. Write a shell script to find a file/s that matches a pattern given as command line argument in the home directory, display the contents of the file and copy the file into the directory ~mydir
b. Write a shell script to list all the files in a directory whose filename is at least 10 characters. (use expr command to check the lenght)

7a. Write a shell script that gets executed displays the message either “Good Morning” or “Good Afternoon” or “Good Evening” depending upon time at which the user logs in.
b Write a shell script that accept a list of filenames as its argument, count and report occurrence of each word that is present in the first argument file on other argument files.

8a. Write a shell script that determine the period for which a specified user is working on system and display appropriate message.
b Write a shell script that reports the logging in of a specified user within one minute after he/she log in. The script automatically terminate if specified user does not log in during a specified period of time.

9a. Write a shell script that accept the file name, starting and ending line number as an argument and display all the lines between the given line number.
b. Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th, a “\" is to be appended as the indication of folding and the processing is to be continued with the residue. The input is to be supplied through a text file created by the user.

10a. Write an awk script that accepts date argument in the form of dd-mm-yy and displays it in the form of month, day and year. The script should check the validity of the argument and in the case of error, display a suitable message.
b. Write an awk script to delete duplicated line from a text file. The order of the original lines must remain unchanged.

11a. Write an awk script to find out total number of books sold in each discipline as well as total book sold using associate array down table as given below.

<table>
<thead>
<tr>
<th>Discipline</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>Electrical</td>
<td>34</td>
</tr>
<tr>
<td>Mechanical</td>
<td>67</td>
</tr>
<tr>
<td>Electrical</td>
<td>80</td>
</tr>
<tr>
<td>Computer Science</td>
<td>43</td>
</tr>
<tr>
<td>Mechanical</td>
<td>65</td>
</tr>
<tr>
<td>Civil</td>
<td>98</td>
</tr>
<tr>
<td>Computer Science</td>
<td>64</td>
</tr>
</tbody>
</table>

b. Write an awk script to compute gross salary of an employee accordingly to rule given below.

If basic salary is < 10000 then HRA=15% of basic & DA=45% of basic
If basic salary is >=10000 then HRA=20% of basic & DA=50% of basic.

Note: In the examination each student picks one question from a lot of all the 11
questions. *Question A & B Not to be included for examination*
1. Create an XHTML page to demonstrate the usage of a. Text Formatting tags, b. Links, c. Images d. Tables

2. Develop and demonstrate the usage of inline and external style sheet using CSS

3. Develop and demonstrate a XHTML file that includes JavaScript script for the following problems:
   a) Input: A number n obtained using prompt
   Output: The first n Fibonacci numbers
   b) Input: A number n obtained using prompt
   Output: A table of numbers from 1 to n and their squares using alert

4. Develop and demonstrate using JavaScript, a XHTML document that displays random numbers (integers).

5. a) Develop and demonstrate, using JavaScript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.

   b) Modify the above program to get the current semester also (restricted to be a number from 1 to 6)

6. a) Develop and demonstrate, using JavaScript script, a XHTML document that contains three images, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.

   b) Modify the above document so that when an image is moved from the top stacking position, it returns to its original position rather than to the bottom.

7. Develop using JavaScript script, an XHTML document that use of onload and onfocus events

8. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Branch, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

   b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.
9. Write a Perl program which demonstrates the usage of scalar variables and arrays

10. Write a Perl program to display various Server information like Server Name, Server Software, Server protocol, CGI Revision etc.

11. Write a Perl program to display a digital clock which displays the current time of the server

12. Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.

13. Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.

14. Write a CGI-Perl program to use a cookie to remember the day of the last login from a user and display it when run

**Note:** In the examination each student picks one question from the lot of all 14 questions.
<table>
<thead>
<tr>
<th>Sub Code</th>
<th>IA Marks</th>
<th>Hrs/Week</th>
<th>Exam Hours</th>
<th>Total Hours</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA21</td>
<td>50</td>
<td>4</td>
<td>3</td>
<td>52</td>
<td>100</td>
</tr>
</tbody>
</table>

**SEMESTER II**  
**Data Structures Using C**

**Introduction to Data Structures**  
10 Hours  
Information and its meaning: Abstract Data Types, Sequences as Value Definitions, ADT for Varying length character Strings, Data Types, Pointers and review of Pointers, Data Structures. Arrays: Array as an ADT, Using One-dimensional Arrays, Implementing One-Dimensional Arrays, Arrays as Parameters, Handling of Character Strings and Character Strings.

**The Stack**  
8 Hours  
Definition and examples, Primitive operations, Example, The stack as an ADT, Representing stacks, Implementing the pop operation, Testing for exceptional conditions, Implementing the push operations, Examples for infix, postfix, and prefix expressions, Basic definition and Examples, Program to evaluate a postfix expression, Converting an expression from infix to postfix, Program to convert an expression from infix to postfix, Applications of Stacks: Expression Evaluations, Recursion etc.

**Recursion**  
4 Hours  
Recursive definition and processes, Factorial function, Multiplication of natural numbers, Fibonacci sequence, Binary search, Properties of recursive definition or algorithm. Binary search, Towers of Hanoi problem.

**Queues and Lists**  
12 Hours  
The queue and its sequential representation, the queue as ADT, Insert operation, Priority queue, Array implementation of a priority queue. Linked lists, Inserting and removing nodes from a list, Linked implementations of stacks, getnode and Freenode operations, Linked implementation of queues, Linked list as a data Structure, Example of list operations, Header nodes, Array implementation of lists, Limitations of array implementation, allocating and freeing dynamic variables, Linked lists using dynamic variables, Non integer and non-homogenous lists, Other list structures: Circular lists, Stack as a circular lists, doubly linked lists, Application of Linked Lists: Stacks, Queues, Double-ended Queues, Priority Queues.

**Sorting**  
8 Hours  
Bubble sort, Quick sort, Selection sort, Tree Sorting: Binary Tree Sort, Heap Sort, Insertion Sorts: Simple Insertion, Shell Sort, Address Calculation Sort, Merge and Radix Sort.

**Searching**  
5 Hours  
Basic Search Techniques: Algorithmic Notations, Sequential searching, Searching an ordered table, Indexed sequential search, Binary search, Interpolation search, Tree searching: Inserting into a Binary Search Tree, Deleting from a binary search tree, Hashing: Resolving hash clashes by open addressing, Choosing a hash Function.

**Binary Trees**  
5 Hours  
Tree traversals, Binary Search Tree and Operations, AVL Tree and Operations, Red-Black Tree, Threaded binary trees and operations.

**Text Books:**  
1. *Data Structures Using C and C++* by Yedidyah Langsam and Moshe J. Augenstein
Reference Books:
Object Oriented Programming Using C++

Sub Code : 13MCA22
IA Marks : 50
Hours/Week : 4
Exam Hours : 3
Total Hours : 52
Exam Marks: 100

Introduction 6 Hours

Modular Programming with Functions 7 Hours
Function Components, argument passing, inline functions, function overloading, function templates, recursive functions.

Classes & Objects 14 Hours
Introduction, Class Specification, Class Objects, access members, defining member functions, data hiding, constructors, destructors, parameterized constructors, static data members, functions, scope resolution operator, passing objects as arguments, returning objects, friend functions & classes, arrays of objects, Dynamic objects – Pointers to objects, Class members, Operator overloading using friend functions such as ++, --, [ ] etc. Class templates.

Inheritance Virtual functions & Polymorphism 12 Hours
Base Class, Inheritance & protected members, protected base class inheritance, inheriting multiple base classes, Constructors, Destructors & Inheritance. Passing parameters to base Class Constructors, Granting access, Virtual base classes, Virtual function -Calling a Virtual function through a base class reference, Virtual attribute is inherited, Virtual functions are hierarchical, pure virtual functions, abstract classes, using Virtual functions, Early & late binding.

I/O Streams 5 Hours
IO Stream basics, output operator <<, input >>, additional I/O operators, overloading the output operator <<, overloading the input operator >>, file input & output, manipulators.

Exception Handling, STL 8 Hours
Exception handling fundamentals, Exception handling options, STL: An overview, containers, vectors, lists, maps.

Text Books:

Reference Book:
4. Object oriented programming with C++, E. Balaguruswamy, TMH.
Introduction: Computer and Operating Systems  6 Hours

Operating System Structures: System Structures  5 Hours
System Components, Operating – System Services, System Calls, System Programs, System Structure, Virtual Machines, System Design and Implementation, System Generation.

Process Management  6 Hours

Mutual Execution and Synchronization  6 Hours
Principles of Concurrency, Mutual Exclusion: Hardware Support, Semaphores, Monitors, Message Passing, Readers/Writes Problem

Deadlock and Starvation  4 Hours
Principles of Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, An Integrated Deadlock Strategy, Dining Philosophers Problem

Memory Management  8 Hours
Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Process Creation, Page Replacement, Allocation of Frames, Thrashing

File – System Interface and Implementation  7 Hours

Secondary Storage, Computer Security  4 Hours

Case study of Linux Operating system:  6 Hours
Text Books

Reference Books
System Software

Subject Code: 13MCA24  
I.A. Marks : 50  
Hours/Week : 04  
Exam Hours: 03  
Total Hours : 52  
Exam Marks : 100

Machine Architecture  
6 Hours

Assemblers  
12 Hours

Loaders and Linkers  
8 Hours

Editors And Debugging Systems  
6 Hours
Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship With Other Parts Of The System, User-Interface Criteria

Macro Processor  
8 Hours

Compilers  
12 Hours

Text Books:
   (Chapters 1.1 to 1.3, 2 (except 2.5.2 and 2.5.3), 3 (except 3.5.2 and 3.5.3), 4 (except 4.4.3), 5.1, 5.2, 5.3 (except 5.3.3, 5.3.4), 5.4)

Reference Books:
Database Management Systems

Subject code : 13MCA25       I.A.Marks  : 50
Hours/Week : 04         Exam Hours : 03
Total Hours : 52        Exam Marks : 100

Unit-1: Introduction            7 Hours
An example; Characteristics of Database approach; Actors on the screen; Workers behind the scene; Advantages of using DBMS approach; A brief history of Database Applications; When not to use a DBMS. Data Models, Schemas and Instances; Three-Schema Architecture and Data Independence; Database Languages and Interfaces; The Database System Environment; Centralized and Client-Server Architectures for DBMSs; Classification of Database Management Systems

Unit-2: Entity-Relationship Model 7 Hours
Using High-Level Conceptual Data Models for Database Design; An Example Database Application; Entity Types, Entity Sets, Attributes and Keys; Relationship Types, Relationship Sets, Roles and Structural Constraints; Weak Entity Types; Refining the ER Design for COMPANY Database; ER Diagrams, Naming Conventions and Design Issues; Relationship Types of Degree Higher than Two, Relational Database Design Using ER- to-Relational Mapping

Unit-3: Relational Model and Relational Algebra 12 Hours
Relational Model Concepts; Relational Model Constraints and Relational Database Schemas; Update Operations, Transactions and Dealing with Constraint Violations; Unary Relational Operations: SELECT and PROJECT; Relational Algebra Operations from Set Theory; Binary Relational Operations: JOIN and DIVISION; Additional Relational Operations; Examples of Queries in Relational Algebra;

Unit-4: SQL 12 Hours
SQL Data Definition and Data Types; Specifying Constraints in SQL; Schema Change Statements in SQL; Basic Queries in SQL; More Complex SQL Queries, Insert, Delete and Update Statements in SQL; Specifying Constraints as Assertions and Triggers; Views (Virtual Tables) in SQL; Additional Features of SQL; Database Programming: Issues and Techniques; Embedded SQL, Dynamic SQL; Database Stored Procedures and SQL / PSM.

Unit-5: Database Design 8 Hours
Informal Design Guidelines for Relation Schemas; Functional Dependencies; Normal Forms Based on Primary Keys; General Definitions of Second and Third Normal Forms; Boyce-Codd Normal Form

Unit-6: PL/SQL 6 Hours
Introduction, Language fundamentals, conditional and sequential control, Iterative processing and loops. Exception handlers, triggers. Functions, procedures. Creating and planning PL/SQL.

Text Books:
Reference Books:

Write a C program to
1. Convert a prefix notation to postfix notation.
2. Evaluate a given postfix expression and its values for the variables.
3. Simulate the working of a circular queue providing the following operations–Insert, Delete and Display.
4. Demonstrate recursion
   a. Calculate GCD and LCM of 3 integer numbers
   b. Solve Towers of Hanoi Problem
   c. Calculate the sum for a given number ‘n’ from 1 to n.
5. Simulate the working of a linked list providing the following operations
   a. Insert at the beginning
   b. Insert at the end
   c. Insert before a given element
   d. Insert at the position
   e. Display
6. Simulate the working of a circular linked list providing the following operations
   a. Delete from the beginning
   b. Delete from the end
   c. Delete a given element
   d. Delete every alternate element
   e. Display
   Insert is mandatory.
7. Simulate the working of a dequeue.
8. Simulate the working of a double linked list to implement stack and queue.
9. Create a binary tree and implement the tree traversal techniques of inorder, preorder and postorder.
10. Implement quick sort.
11. Implement Heap sort.
12. Implement the search techniques of
    a. Linear Search
    b. Binary Search
13. Write a Program to
    a) Create AVL Tree
    b) Insert element to AVL tree
    c) Find the height of the AVL tree
Database Laboratory

Sub code       : 13MCA27          IA Marks: 50
Hours/week : 3         Exam Hours: 3
Total hours : 42                        Exam Marks: 50

Instructions for the Exercises:

1. Draw ER diagram based on given scenario with various Constraints.
2. Create Relational Database Schema based on the above scenario using Mapping Rules.
3. Perform the given queries using any RDBMS Environment.
4. Suitable tuples have to be entered so that queries are executed correctly
5. The results of the queries may be displayed directly

Exercise: 1

Notown Records has decided to store information about musicians who perform on its albums. Each musician that records at Notown has an SSN, a name, an address, and a phone number. No musician has more than one phone. Each instrument used in songs recorded at Notown has a unique identification number, a name (e.g., guitar, synthesizer, flute) and a musical key (e.g., C, B-flat, E-flat). Each album recorded on the Notown label has a unique identification number, a title, a copyright date and a format (e.g., CD or MC). Each song recorded at Notown has a title and an author. Each musician may play several instruments, and a given instrument may be played by several musicians. Each album has a number of songs on it, but no song may appear on more than one album. Each song is performed by one or more musicians, and a musician may perform a number of songs. Each album has exactly one musician who acts as its producer. A musician may produce several albums.

Queries

a) List musician name, title of the song which he has played, the album in which song has occurred.
b) List the details of songs which are performed by more than 3 musicians.
c) List the different instruments played by the musicians and the average number of musicians who play the instrument.
d) Retrieve album title produced by the producer who plays guitar as well as flute and has produced no of songs greater than the average songs produced by all producers.
e) List the details of musicians who can play all the instruments present.

Exercise: 2

Professors have a PROFID, a name, an age, a rank, and a research specialty. Projects have a project number, a sponsor name (e.g. UGC/AICTE/...), a starting date, an ending date, and a budget. Graduate students have an USN, a name, an age, and a degree program (e.g. MCA/MPhil/BE/ME ..). Each project is managed exactly by one professor (known as the project's principal investigator). Each project is worked on by one or more professors (known as the project's co-investigators). Professors can manage/work on multiple projects. Each project is worked on by one or more graduate students (known as the project's research assistants). Graduate students can work on multiple projects. Each professor can supervise many students. A student who is working on a project can be supervised by only one professor.
Queries

a) Retrieve the names of all professors who do not have an ongoing project of more than 1 lakh.
b) Retrieve the names of all graduate students along with their professors under whom they work and project sponsor.
c) List the professors and sum of the budget of their projects started after 2005 but ended in 2010.
d) List the names of professors who has a total worth of project greater than the average budget of projects sanctioned

e) List the professors who work on all the projects.

Exercise: 3

A bank has many branches and a large number of customers. Bank is identified by its code. Other details like name, address and phone for each bank are also stored. Each branch is identified by its bank. Branch has name, address and phone. A customer can open different kinds of accounts with the branches. An account can belong to more than one customer. Customers are identified by their SSN, name, address and phone number. Age is used as a factor to check whether customer is a major. There are different types of loans, each identified by a loan number. A customer can take more than one type of loan and a loan can be given to more than one customer. Loans have a duration and interest rate. Make suitable assumptions and use them in showing maximum and minimum cardinality ratios.

Queries:

a) List the details of customers who have joint account and also have at least one loan.
b) List the details of the branch which has given maximum loan.
c) List the details of saving accounts opened in the SBI branches located at Bangalore.
d) List the name of branch along with its bank name and total amount of loan given by it.
e) Retrieve the names of customers who have accounts in all the branches located in a specific city.

Exercise: 4

Patients are identified by an SSN, and their names, addresses, and ages must be recorded. Doctors are identified by an SSN. For each doctor, the name, specialty, and years of experience must be recorded. Each pharmaceutical company is identified by name; it has an address and one phone number. For each drug, the trade name and formula must be recorded. Each drug is sold by a given pharmaceutical company, and the trade name identifies a drug uniquely from among the products of that company. Each pharmacy has a name, address, and phone number. Each patient is checked up by some doctor. Every doctor has at least one patient. Each pharmacy sells several drugs and has a price for each. A drug could be sold at several pharmacies, and the price could vary from one pharmacy to another. Doctors prescribe drugs for patients. A doctor could prescribe one or more drugs for several patients, and a patient could obtain prescriptions from several doctors. Each prescription has a date and a quantity associated with it. Pharmaceutical companies have long-term contracts with pharmacies. A pharmaceutical company can contract with several pharmacies, and a pharmacy can contract with several pharmaceutical companies. For each contract, you have to store a start date, an end date, supervisor and the text of the contract.
Queries:

a) List the details of patients who are 20 years old and have been checked by eye-specialist.
b) List the details of doctors who have given the prescription to more than 20 patients in year 2013.
c) List the details of pharmaceutical companies who supply drug to more than 10 pharmacies in the same city where company is located.
d) List the details of drug supplied by only one pharmaceutical company.
e) List the details of drug supplied by all pharmaceutical companies.

Exercise; 5

Data requirements of movie industry are captured. Each movie is identified by title and year of release. Each movie has length in minutes and classified under one genres (like action, horror etc.). Each movie has a plot outline. Production companies are identified by name and each has an address. A production company produces one or more movies. Actors are identified by id. Other details like name and date of birth of actors are also stored. Each actor acts in one or more movies. Each actor has a role in movie. Directors are identified by id. Other details like name and date of birth of directors are also stored. Each director directs one or more movies. Each movie has one or more actors and one or more directors and is produced by a production company.

Queries:

a) List the details of horror movies released in 2012 and directed by more than 2 directors.
b) List the details of actors who acted in movies having same titles but released before 2000 and after 2010.
c) List the details of production companies producing maximum movies.
d) List the details of movies where director and actor have same date of birth.
e) Retrieve the names of directors directed all the movies produced by any one production company.

Note: In the examination each student picks one question from a lot of 5 exercises.
1. Define a STUDENT class with USN, Name, and Marks in 3 tests of a subject. Declare an array of 10 STUDENT objects. Using appropriate functions, find the average of the two better marks for each student. Print the USN, Name and the average marks of all the students.

2. Write a C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a complex number:
   (i) ADD (a, s2) – where ‘a’ is an integer (real part) and s2 is a complex number
   (ii) ADD (s1, s2) – where s1 and s2 are complex numbers

3. Write a C++ program for scalar multiplication of two vectors using operator overloading.

4. Write a C++ program to create a template function for Bubble Sort and demonstrate sorting of integers and doubles.

5. Write a C++ program to create a class called LIST (linked list) with member functions to insert an element at the front and delete an element from the front of the list. Demonstrate all the functions after creating a LIST object.

6. Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators ‘+’ and ‘--’:
   (i) s1 = s1 + element; where s1 is an object of the class STACK and element is an integer to be pushed on the top of the stack
   (ii) s1 = --s1 ; where s1 is an object of the class STACK. ‘--’ operator pops the element. Handle the STACK empty and full conditions. Also display the contents of the stack after each operation, by overloading the << operator.

7. Create a class called MATRIX using two-dimensional array of integers. Implement the following operations by overloading the operator ++ which checks the compatibility of two matrices to be added and subtracted. Perform the addition and subtraction by overloading the + and – operators respectively. Display the results by overloading the operator <<. If (m1==m2) then m3 = m1+m2 and m4 = m1-m2 else display error.

8. Write a C++ program to create a class called OCTAL which has the characteristics of an octal number. Implement the following operations by writing an appropriate constructor and an overloaded operator +.
   (i) OCTAL h = x; where x is an integer.
   (ii) int y = h + k; where h is an OCTAL object and k is an integer
   Display the OCTAL result by overloading the operator <<. Also display the values of h and y.

9. Write a C++ program to create a class template called QUEUE with member functions to add an element and to delete an element from the queue. Using the member functions, implement a queue of integers and double. Demonstrate the operations by displaying the
contents of the queue after every operation.

10. Define a class SET with Data members: array of int, int variable to indicate number of elements in a SET object; and Member functions: to read element of a SET object, to print elements of a SET object, to find union of 2 objects of SET using operator overloading (S3=S1+S2), to find intersection of 2 objects of SET using operator overloading (S4= S1*S2). S1, S2, S3 and S4 are objects of SET. Use this class in a main function to show the above operations.

11. Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UG STUDENT and PG STUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students. Find the semester-wise average age for all UG and PG students separately.

12. Write a C++ program to create a class called STRING and implement the following operations. Display the results after every operation by overloading the operator <<.
   (i) STRING s1 = “VTU”
   (ii) STRING s2 = “BELGAUM”
   (iii) STRING s3 = s1 + s2 (Use copy constructor)

13. Define a base class STACK1 which performs only push, pop, display operations. Override the above operations through a derived class STACK2 which takes care of STACK FULL & STACK EMPTY situations. Show how the objects of these classes use the above functions in a main function.

14. Create an abstract base class EMPLOYEE with data members: Name, EmpID and BasicSal and a pure virtual function Cal_Sal(). Create two derived classes MANAGER (with data members: DA and HRA) and SALESMAN (with data members: DA, HRA and TA). Write appropriate constructors and member functions to initialize the data, read and write the data and to calculate the net salary. The main() function should create array of base class pointers/references to invoke overridden functions and hence to implement run-time polymorphism.

Note: In the examination each student has to pick one question from a lot of all the 14 questions.
SEMESTER III

Computer Networks

<table>
<thead>
<tr>
<th>Subject Code: 13MCA31</th>
<th>IA.Marks : 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hours/Week : 4 Hrs.</td>
<td>Exam Duration : 3 Hrs</td>
</tr>
<tr>
<td>Total Hours : 52 Hrs.</td>
<td>Examination Marks : 100</td>
</tr>
</tbody>
</table>

Introduction to Computer Networks

Physical Layer
- Data Transmission Concepts, Analog and Digital Data Transmission, Transmission Impairments and Channel Capacity, Guided and Wireless transmission, communication media, Digital modulation techniques (FDMA, TDMA, CDMA) and mobile telephone systems (1G, 2G, 3G and 4G).

Data Link layer
- Error Detection and Correction Codes, Data Link Protocols and Sliding window protocols.

Medium Access Sub Layer
- Multiple access protocols and Examples: Ethernet, Wireless LAN, Broadband Wireless and bluetooth, Data Link Layer Switching.

Network Layer
- Network Layer Design issues, Routing algorithms, Congestion Control Algorithms, Quality of Service, Internetworking and The Network Layer in the Internet

The Transport Layer

The application Layer
- DNS, Email, WWW, Streaming audio and Video and Content Delivery

Text Books
   - Chapter 1, 2.2, 2.3, 2.5, 2.7, 3.1, 3.2, 3.3, 3.4, 4.2, 4.3, 4.4, 4.5, 4.6, 4.8 Chapter 5, Chapter 6 (excluding 6.7)

Reference Books
2. [http://www.ietf.org/rfc.htm](http://www.ietf.org/rfc.htm) relevant RFC document could be used to get more detailed information about any of the concepts prescribed in the syllabus like RFC 2460 can be referred to get a detailed information about IPV6
Programming using JAVA

Subject Code : 13MCA32  
I.A. Marks: 50
Hours/Week : 4  
Exam Hours : 3
Total Hours : 52  
Exam Marks : 100

Java Programming Fundamentals  
2 Hours

Introducing Data Types and Operators  
1 Hour
Java’s Primitive Types, Literals, A Closer Look at Variables, The Scope and Lifetime of Variables, operators, Shorthand Assignments, Type conversion in Assignments, Using Cast, Operator Precedence, Expressions.

Program Control Statements  
1 Hour
Input characters from the Keyword, if statement, Nested ifs, if-else-if Ladder, Switch Statement, Nested switch statements, for Loop, Enhanced for Loop, While Loop, do-while Loop, Use break, Use continue, Nested Loops

Introducing Classes, Objects and Methods  
3 Hours
Class Fundamentals, How Objects are Created, Reference Variables and Assignment, Methods, Returning from a Method, Returning Value, Using Parameters, Constructors, Parameterized Constructors, The new operator Revisited, Garbage Collection and Finalizers, The this Keyword.

More Data Types and Operators  
2 Hours
Arrays, Multidimensional Arrays, Alternative Array Declaration Syntax, Assigning Array References, Using the Length Member, The For-Each Style for Loop, Strings, The Bitwise operators.

String Handling  
2 Hours
String Fundamentals, The String Constructors, Three String-Related Language Features, The Length() Method, Obtaining the characters within a string, String comparison, using indexOf() and lastIndexOf(), Changing the case of characters within a string, StringBuffer and String Builder.

A Closer Look at Methods and Classes  
3 Hours
Controlling Access to Class Members, Pass Objects to Methods, How Arguments are passed, Returning Objects, Method Overloading, Overloading Constructors, Recursion, Understanding Static, Introducing Nested and Inner Classes, Varargs: Variable-Length Arguments.

Inheritance  
6 Hours
Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call Superclass constructors, Using super to Access Superclass Members, Creating a Multilevel Hierarchy, When are Constructors Executed, Superclass References and Subclass Objects, Method Overriding, Overridden Methods support polymorphism, Why Overridden Methods, Using Abstract Classes, Using final, The Object Class.
Interfaces 3 Hours
Interface Fundamentals, Creating an Interface, Implementing an Interface, Using Interface References, Implementing Multiple Interfaces, Constants in Interfaces, Interfaces can be extended, Nested Interfaces, Final Thoughts on Interfaces.

Packages 3 Hours
Package Fundamentals, Packages and Member Access, Importing Packages, Static Import

Exception Handling 4 Hours
The Exception Hierarchy, Exception Handling Fundamentals, The Consequences of an Uncaught Exception, Exceptions Enable you to handle errors gracefully, using Multiple catch clauses, Catching subclass Exceptions, try blocks can be nested, Throwing an Exception, A Closer look at Throwable, using finally, using throws, Java’s Built-in Exceptions, New Exception features added by JDK 7, Creating Exception Subclasses.

Multithreaded Programming 4 Hours
Multithreading fundamentals, The Thread Class and Runnable Interface, Creating Thread, Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities, Synchronization, using Synchronization Methods, The Synchronized Statement, Thread Communication using notify(), wait() and notify All(), suspending, Resuming and stopping Threads.

Enumerations, Auto boxing and Annotations 4 Hours
Enumerations, Java Enumeration are class types, The Values () and Valueof () Methods, Constructors, methods, instance variables and enumerations, Auto boxing, Annotations (metadata)

Generics 2 Hours
Generics Fundamentals Bounded Types, Generic Methods, Generic Constructors, Some Generic Restrictions.

Applets 2 Hours
Applet basics, A complete Applet Skeleton, Applet Initialization and Termination, A key Aspect of an Applet Architecture, Requesting Repainting, using the status window, Passing parameters to Applets.

Swing Fundamentals 4 Hours
The origin and Design philosophy of swing, Components and containers, Layout managers, A first simple swing Example, Event Handling, Exploring Swing Controls-JLabel and ImageIcon, The Swing Buttons, Trees.

Networking with Java.net 3 Hours

Exploring Collection Framework 3 Hours
Collections Overview, The Collection Interfaces, The collection Classes. The Arrays Class.

Text Books:
1. J
ava Fundamentals, A comprehensive Introduction by Herbert Schildt, Dale Skrien. Tata McGraw Hill Edition 2013. (Chapters: 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,17,18,22,23,24,25,26)

Reference Books:
2. Programming in JAVA2 by Dr K Somasundaram, Jaico publications
5. Java 2 Essentials, Cay Hortsmann, second edition, Wiley
Software Engineering

Subject Code: 13MCA33
I.A. Marks : 50
Hours/Week: 4
Exam Hours : 3
Total Hours : 52
Exam Marks : 100

1. Overview
   Introduction: Professional Software Development Attributes of good software, software engineering diversity, IEEE/ACM code of software engineering ethics, case studies

2. Software Process & Agile Software Development

3. Requirements Engineering
   Functional and non-functional requirements, The software requirements document, Requirements specification, Requirements engineering processes, Requirement elicitation and analysis, Requirements validation, Requirements management

   Context models, Interaction models, Structural models, Behavioral models, Model-driven engineering, Software architecture: the role of software architecture, architectural views, component and connector view, Architectural styles for C&C view, Documenting architectural design. Design: Design concepts, Function oriented design, detailed design, verification, matrix (Complexity matrix for function oriented design),

5. Component-based software engineering
   Components and component model, CBSE process, Component composition

6. Distributed Software engineering
   Distributed system issues, Client-server computing, Architectural patterns for distributed systems, Software as a service.

7. Planning a software Project
   Process planning, Effort estimation, Project scheduling and staffing, Software configuration management plan, Quality plan, Risk Management, Project monitoring plan.

8. Software Testing
   Testing fundamentals, Black-box testing, White-box testing, Testing process

Text Books:
1. Ian Sommerville: Software Engineering, 9th edition, Person Education Ltd, 2011. (Chapters: 1, 2, 3, 4, 5, 17, 18)

Reference Books:
Software Engineering architectural driven software developments by R Schmidt
Computer Graphics

Subject Code : 13MCA34
Total Hours : 52
Hours/Week : 4
I.A. Marks: 50
Exam Hours: 3
Exam Marks: 100

Graphics Output Primitives and Attributes
Introduction to open GL, Coordinate reference frames, Specifying two dimensional world coordinate reference frame in Open GL, Open GL point functions, Open GL line functions, Line drawing algorithms, Circle generation algorithms, Ellipse generation algorithms, Fill area primitives, Polygon fill areas, OpenGL polygon fill area functions, General scan line polygon fill algorithm, Fill methods for areas with irregular boundaries, Open GL fill area attribute functions

Two – Dimensional and Three - Dimensional Geometric Transformations
Basic two dimensional geometric transformations, Matrix representations and homogeneous coordinates, Inverse transformations, Two dimensional composite transformations, Other two dimensional transformations, Three dimensional Translation, Rotation, Scaling, Other three dimensional transformations, Affine transformations, OpenGL geometric transformation functions

Two Dimensional Viewing
The two dimensional viewing, Clipping window, Normalization and viewport transformations, Clipping algorithms, Two dimensional point clipping, Two dimensional line clipping algorithms, Polygon fill area clipping, Curve clipping, Text clipping

Three Dimensional Viewing
The three dimensional viewing concepts, Three dimensional viewing pipeline, Three dimensional viewing coordinate parameters, Transformation from world to viewing coordinates, Projection transformations, Orthogonal projections, Oblique parallel projections, Perspective projections, The viewport transformation and three dimensional screen coordinates

Curves and Computer Animation
Bezier spline curves, Raster methods for computer animation, Design of animation sequences, Traditional animation techniques, General computer animation functions

Text book:

Reference Books:
UNIX System Programming

**Subject Code:** 13MCA351

**I.A. Marks:** 50

**Hours/Week:** 4

**Exam Hours:** 03

**Total Hours:** 52

**Exam Marks:** 100

### Introduction

6 Hours


### UNIX Files

6 Hours


### UNIX File APIs

7 Hours

General File APIs, File and Record Locking, Directory File APIs, Device File APIs, FIFO File APIs, Symbolic Link File APIs, General File Class, regfile Class for Regular Files, dirfile Class for Directory Files, FIFO File Class, Device File Class, Symbolic Link File Class, File Listing Program.

### UNIX Processes

7 Hours


### Process Control

7 Hours


### Signals and Daemon Processes

7 Hours


### Interprocess Communication

6 Hours

Introduction; Pipes, popen, pclose Functions; Coprocesses; FIFOs; XSI IPC; Message Queues; Semaphores

### Network IPC: Sockets

6 Hours

Introduction; Socket Descriptors; Addressing; Connection establishment; Data transfer; Socket options; Out-of-band data; Nonblocking and asynchronous I/O.
Text Books:

1. Terrence Chan: Unix System Programming Using C++, Prentice-Hall of India / Pearson Education, 1999. (Chapters 1, 5, 6, 7, 8, 9)

Over view of Storage and Indexing, Disks and Files  
7 Hours
Data on external storage; File organizations and indexing; Index data structures; Comparison of file organizations; Indexes and performance tuning; Memory hierarchy; RAID; Disk space management; Buffer manager; Files of records; Page formats and record formats.

Transaction Management  
6 Hours
Introduction to Transaction Processing; Transaction and System Concepts; Desirable Properties of Transactions; Characterizing Schedules based on Recoverability; Characterizing Schedules based on Serializability; Two-Phase Locking Techniques for Concurrency Control; Concurrency Control based on Timestamp Ordering; Multiversion Concurrency Control Techniques; Validation Concurrency Control Techniques; Granularity of Data Items and Multiple Granularity Locking; Recovery Concepts; Recovery Techniques based on Deferred Update; Recovery Techniques based on Immediate Update; Shadow Paging; The ARIES Recovery Algorithms; Recovery in Multidatabase Systems; Database Backup and Recovery from Catastrophic Failures.

Tree Structured Indexing  
7 Hours
Intuition for tree indexes; Indexed sequential access method; B+trees, Search, Insert, Delete, Duplicates, B+tress in practice

Hash-Based Indexing  
4 Hours
Static hashing, Extendible hashing, Linear hashing, comparisons

Overview of Query Evaluation, External Sorting  
5 Hours
The system catalog, Introduction to operator evaluation; Algorithm for relational operations; Introduction to query optimization; Alternative plans; A motivating example; what a typical optimizer does. When does a DBMS sort data? A simple two-way merge sort; External merge sort

Evaluating Relational Operators  
4 Hours
The Selection operation; General selection conditions; The Projection operation; The Join operation; The Set operations; Aggregate operations; The impact of buffering.

A Typical Relational Query Optimizer  
7 Hours
Translating SQL queries in to Relational Algebra; Estimating the cost of a plan; Relational algebra equivalences; Enumeration of alternative plans; Nested sub-queries; other approaches to query optimization.

Physical Database Design and Tuning  
6 Hours
Introduction; Guidelines for index selection, examples; Clustering and indexing; Indexes that enable index-only plans, Tools to assist in index selection; Overview of database tuning; Choices in tuning the conceptual schema; Choices in tuning queries and views; Impact of concurrency; DBMS benchmarking.
More Recent Applications 6 Hours
Mobile databases; Multimedia databases; Geographical Information Systems; Genome data management.

Text Books:

Reference Books:
Basics of MIS and E-Commerce

Subject Code: 13MCA353 I.A. Marks : 50
Hours/Week: 4 Exam Hours : 3
Total Hours: 52 Exam Marks : 100

Information and Knowledge 4 Hours
Information concepts, classification of information, methods of data and information collection, value of information, information: A quality product, General model of a human as information processor, Knowledge,

Introduction of MIS 4 Hours
MIS: Concept, Definition, Role of the MIS, Impact of MIS, MIS and the user, Management as a control system, MIS support to the management, Management effectiveness and MIS, Organization as system. MIS: organization effectiveness

Decision Making and DSS 4 Hours
Decision making concepts; decision making process, decision-making by analytical modeling, Behavioral concepts in decision making, organizational decision-making, Decision structure, DSS components, Management reporting alternatives.

Electronic Business systems 5 Hours
Enterprise business system – Introduction, cross-functional enterprise applications, real world case, Functional business system, - Introduction, marketing systems, sales force automation, CIM, HRM, online accounting system, Customer relationship management, ERP, Supply chain management (real world cases for the above)

Client Server Architecture and E-business Technology 8 Hours

E-Commerce Introduction 9 Hours
Course overview: Introduction to e-commerce, E-commerce Business Models and Concepts, E-Commerce Infrastructure: The Internet and World Wide Web, Web design, JavaScript Internet Information Server (IIS); Personal Web Server (PWS),

E-Commerce techniques and Issues 9 Hours
Introduction to Active Server Pages (ASP), Building an E-Commerce Web Site, E-Commerce Payment Systems, E-Commerce Marketing Techniques, Building product catalogue, Search product catalogue, Web Spider and search agent, Ethical, Social and Political Issues in E-Commerce

Internet Communication 9 Hours

Text Books:

Reference Books:
4. Rahul De, Managing Information Systems in Business, Government and Society, Wiley India.
Introduction and Overview of the OR Modeling Approach

The origin of OR, the nature of OR, the impact of OR, defining the problem and gathering data, Formulating a mathematical model, deriving solutions from the model, testing the model, preparing to apply the model, implementation.

Introduction to Linear Programming

Formulation of linear programming problem (LPP), examples, Graphical solution, the LP Model, Special cases of Graphical method, assumptions of Linear Programming (LP), additional example

Solving LPP - the Simplex Method

The essence of the simplex method, setting up the simplex method, algebra of the simplex method, the simplex method in tabular form, special cases in the simplex method, tie breaking in the simplex method, adopting to other model forms (Two Phase method, Big-M method), post optimality analysis.

Duality Theory and Sensitivity Analysis

The essence of duality theory, economic interpretation of duality, primal dual relationship, adapting to other primal forms, the role of duality in sensitive analysis, the dual simplex method

Transportation and Assignment Problems

The transportation problem, a stream line simplex method for the transportation problem, the assignment problem, a special algorithm for the assignment problem

PERT and CPM

Network representation, Critical path (CPM) computations and PERT networks.

Game Theory

The formulation of two persons, zero sum games, solving simple games- a prototype example, games with mixed strategies, graphical solution procedure, solving by linear programming, extensions

Text Books:


Reference Books:

2. Theory and Problems of Operations Research, Richard Bronson and
Principles of User Interface Design

Subject Code: 13MCA355
L.A. Marks : 50
Hours/Week: 4
Exam Hours : 3
Total Hours: 52
Exam Marks : 100

Introduction 8 Hours
Usability of Interactive Systems: Introduction, Usability Goals and Measures, Usability Motivation, Universal Usability, Goals for our profession. Guideline, principles, and theories: Introduction, Guidelines, principles, Theories,

Development Processes 6 Hours

Evaluating Interface Design 6 Hours
Introduction, Expert Reviews, Usability Testing and Laboratories, Survey Instruments, Acceptance tests, Evaluation during Active Use, Controlled Psychologically Oriented Experiments

Interaction Styles 8 Hours
Direct Manipulation and Virtual Environments: Introduction, Examples of Direct Manipulation, Discussion of direct manipulation, 3D Interfaces, Tele-operation, Virtual and Augmented Reality
Menu Selection, Form Filling and Dialog Boxes: Introduction, Task-Related Menu Organization, Single Menus, Combination of Multiple Menus, Content Organization, Fast Movement Through Menus, Data Entry With Menus, Form Filling, Dialog Boxes and Alternatives, Audio Menus and Menus for Small Displays

Command and Natural Languages 6 Hours
Introduction, Command-organization functionality strategies and structure, Naming and Abbreviations, Natural Language in computing.
Interaction Devices: Introduction, Keyboards and Keypads, Pointing Devices, Speech and Auditory interfaces, Displays-Small and Large

Design Issues 6 Hours
Quality of Service: Introduction, Models of Response-Time Impacts, Expectations and Attitudes, User Productivity, Variability in Response time, Frustrating Experiences
Balancing Function and Fashion: Introduction, Error Messages, Non-anthropomorphic Design, Display design, web page design, Window Design, Color

User Documentation and Online Help : 6 Hours
Introduction, Online versus paper documentation, Reading from paper versus Displays, Shaping the content of the Manuals, Accessing the Documentation, Online Tutorials and animated demonstrations, Online Communities for User Assistance, The Development Process.

Information Search and Visualization 6 Hours
Introduction, Search in Textual Documents and Database Querying, Multimedia document searches, Advanced filtering and Search Interfaces,
Information Visualization : Introduction, Data type by task taxonomy, Challenges for information visualization.
TextBooks


Reference Books

### Probability, Statistics & Numerical Techniques

**Subject Code:** 13MCA356  
**I.A. Marks:** 50  
**Hours/Week:** 4  
**Exam Hours:** 3  
**Total Hours:** 52  
**Exam Marks:** 100

**Introduction**

**Random Variables**
Introduction, Random variables types, functions of random variables, Probability mass functions, The Probability distribution functions, cumulative distribution function, expected values of x, moments, moment generating function, Discrete Distributions, binomial distribution, Poisson distribution, Geometric distribution, continuous distribution, normal distribution, exponential distribution

**Regression and Analysis of Variance**

**Numerical methods for solving algebraic transcendental equations**
Introduction, bisection, Newton’s Raphson

**Matrices**
Elementary row operation, Rank of a matrix, consistency of system of linear equations  
Solutions of system linear equations – Gauss elimination, Gauss-Seidel iterative method

**Numerical integration**
Trapezoidal rule, Simpson’s 1/3 rd rule, Simpson’s 3/8 th rule

**Text Books**

**Reference Books**
Java Programming Laboratory

Subject Code: 13MCA36        I.A. Marks : 50
Hours/Week: 3               Exam Hours : 3
Total Hours: 42             Exam Marks : 50

1. a. Write a JAVA Program to demonstrate Constructor Overloading and Method Overloading.
   b. Write a JAVA Program to implement Inner class and demonstrate its Access protection.

2. Write a program in Java for String handling which performs the following:
   i) Checks the capacity of StringBuffer objects.
   ii) Reverses the contents of a string given on console and converts the resultant string in upper case.
   iii) Reads a string from console and appends it to the resultant string of ii.

3. a. Write a JAVA Program to demonstrate Inheritance.
   b. Simple Program on Java for the implementation of Multiple inheritance using interfaces to calculate the area of a rectangle and triangle.

4. Write a JAVA program which has
   i. A Class called Account that creates account with 500Rs minimum balance, a deposit() method to deposit amount, a withdraw() method to withdraw amount and also throws LessBalanceException if an account holder tries to withdraw money which makes the balance become less than 500Rs.
   ii. A Class called LessBalanceException which returns the statement that says withdraw amount (_______ Rs) is not valid.
   iii. A Class which creates 2 accounts, both account deposit money and one account tries to withdraw more money which generates a LessBalanceException take appropriate action for the same.

5. Write a JAVA program using Synchronized Threads, which demonstrates Producer Consumer concept.

6. Write a JAVA program to implement a Queue using user defined Exception Handling (also make use of throw, throws.).

7. Complete the following:
   1. Create a package named shape.
   2. Create some classes in the package representing some common shapes like Square, Triangle, and Circle.
   3. Import and compile these classes in other program.

8. Write a JAVA Program
   a. Create an enumeration Day of Week with seven values SUNDAY through SATURDAY. Add a method is Workday( ) to the DayofWeek class that returns true if the value on which it is called is MONDAY through FRIDAY. For example, the call DayOfWeek.SUNDAY.isWorkDay ( ) returns false.

9. Write a JAVA program which has
   i. A Interface class for Stack Operations
   ii. A Class that implements the Stack Interface and creates a fixed length Stack.
   iii. A Class that implements the Stack Interface and creates a Dynamic length Stack.
   iv. A Class that uses both the above Stacks through Interface reference and does the Stack operations that demonstrates the runtime binding.

10. Write a JAVA program to print a chessboard pattern.
11. Write a JAVA Program which uses FileInputStream / FileOutPutStream Classes.

12. Write JAVA programs which demonstrates utilities of LinkedList Class.

13. Write a JAVA program which uses Datagram Socket for Client Server Communication.

14. Write a JAVA applet program, which handles keyboard event.

*Note: In the examination each student should do one question out of the above 14 questions*
PART -A
1. Write a program to create a chess board using DDA line algorithm
2. Write a program to implement Bresenham’s line drawing algorithm with all values of slopes
3. Write a program to implement Midpoint circle generation algorithm
4. Write a program to create a wireframe model of globe using equation of ellipse
5. Write a program to create and fill the two dimensional object by using boundary fill algorithm
6. Write a program to create (without using built in function) a cube by implementing translation algorithm by translating along 1. X-axis, 2. Y-axis and 3. X and Y plane
7. Write a program to create (without using built in function) and rotate (1. given an angle 2. Around x and y axis) a triangle by implementing rotation algorithm.
8. Write a program to create (without using built in function) a triangle by implementing scaling algorithm by zooming/un-zooming along 1. X-axis, 2. Y-axis and 3. X and Y plane
9. Write a program to create (without using built in function) a Cube by implementing reflection algorithm 1. X-axis, 2. Y-axis
10. Write a program to create (without using built in function) a square by implementing shear algorithm along 1. X-axis, 2. Y-axis
11. Write a program to animate a flag using Bezier Curve algorithm
12. Write a program to clip lines using Liang-Barsky algorithm

PART –B
1. Develop different chart options with the given inputs by applying DDA algorithm
2. Develop different line styles using Bresenham’s algorithm
3. Develop different circular patterns using midpoint circle generation algorithm
4. Animate cube and globe with given attributes
5. Develop a screen saver with curves with given attributes
6. Develop a screen saver with text with given attributes
7. Develop a screen saver with 2D objects
8. Develop a screen saver with bouncing of 3D objects
10. Animate a bicycle / car with given attributes

Note:
Students Should Complete All Programs from Part-A and Any Two Programs from Part – B Using OpenGL. Consider all types attributes like color, thickness, styles, font, background, speed etc while doing Part - B.

In the examination each student picks one question from the lot of all 12 questions from Part – A and demonstrate any program from Part-B

Online reference:
1. Write a TCL script to simulate the network described below

Consider a small network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 is at the center. Node n0 is a TCP source, which transmits packets to node n3 (a TCP sink) through the node n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration of the simulation time is 10 seconds.
Write a Tcl script (in the text area below) to simulate this scenario. Once you have written your simulation script, click on the "Run" button to execute it. You can verify your code by clicking on the "View Solution" button at the bottom of the page.

2. Write a TCL script to simulate a file transfer with ns2

Consider a client and a server. The server is running a FTP application (over TCP). The client sends a request to download a file of size 10 MB from the server. Write a script to simulate this scenario. Let node #0 be the server and node #1 be the client. TCP packet size is 1500 B. Assume typical values for other parameters.

**Note:** This simulation require transfer of a **fixed** size file. Therefore, time required for the transfer would be constant for a given bandwidth of a link. To verify this, determine the time that would roughly be required for the transfer. Then look at the bottom of the trace file and verify whether there is any transmission beyond the time calculated.
To verify that the client has downloaded the entire file, plot the "Bytes Received " curve for node #1. The y-axis is in Kbits. Convert it to MB and verify whether it approximates the specified file size. TCP headers would effectively increase the count of received bytes at node # 1.

3. Setting up a local area network with ns2

In this exercise you will be simulating a CSMA/CD based LAN with ns2. Consider the LAN with seven nodes to be an isolated one i.e. not connected to the Internet. Node # 0 in the LAN act as a UDP traffic source, and node # 6 is the destination node. Assume CBR traffic to be flowing between the nodes. The simulation lasts for 25 seconds.
In Ethernet a packet is broadcasted in the shared medium, and only the destination node accepts the packet. Other nodes simply drop it. What should be the number of hops a packet from node # 0 to node # 6 travel? Verify this from the "Hop Count" plot.

**Additional Task:**
Suppose the above LAN is to be connected to the Internet. Add node # 7 into the network so that it act as the gateway. Connect node # 0 and # 7 with a 1 Mb wired link. Move the UDP source to node # 7. How the hop count should get affected in this case? Verify from the plot.
4. Simulating link errors

Consider the following network diagram

Here node # 2 act as a router. Any traffic to or from the LAN passes through it. Consider node # 1 running a FTP server, and node # 5 is downloading a file of size 4 MB. However, the link between node # 2 and # 3 is fault. It drops packets with a fixed probability of 0.2. Implement a link error model to reflect this.

It may be noted here that the file download time will be more than the we had in exercise # 2 of experiment # 1. Try different values of the simulation time to ensure that the file has been entirely transferred. Is the plot of bytes received a linear curve or non-linear? Why?

Presence of link errors cause one or more packets to be retransmitted. Verify this from the "Packet Retransmissions" plot.

5. Measuring Network Performances

Bottleneck in the network

Consider a dumbbell topology with eight nodes as shown as in the following figure. Consider nodes# 2 and 3 to be two routers connecting two different networks. When the bandwidth of the link 2-3 is much lower than the sum of bandwidths of the other links in the network, it act as a bottleneck.
Assume node # 0 running a FTP application (over TCP) and sending data to node # 6. Node # 1 is sending CBR data node # 7. Assume all the links except 2-3 has a bandwidth of 1 Mb, propagation delay of 10ms and queue type as DropTail. (All are duplex links).

Tasks:

• The link 2-3 has a propagation delay of 10 ms. Vary it's bandwidth from 0.5 Mb to 2.5 Mb in steps of 0.25Mb.
• Compute the throughput for node # 3 in each case
• Plot the throughput vs. bandwidth data in the "Custom Plot" section below

Based on the above plots, suggest what should be the recommended bandwidth of the link 2-3. Now, plot the end-to-end delay between nodes 0 and 6 for the above chosen values of link 2-3 bandwidth. Revisit your previous answer (i.e. optimum bandwidth of link 2-3) based on these graphs.

Measuring Network Performances
6. Bandwidth sharing between TCP and UDP
Consider the dumbbell topology from our previous exercise:

Node # 0 is a TCP source, and the corresponding sink is at node # 6. Node # 1 is a UDP source (CBR traffic) with a null agent attached to node # 7. These two traffic flows through the common link 2-3. The aim of this exercise is to examine how TCP and UDP share the bandwidth between themselves when the rate of CBR traffic is changed.
Set the TCP packet size to 1460 B. The UDP and CBR packet sizes are 1500 B. All the links in the network have same bandwidths (say, 4 Mb), delay and queue types.
**Part 1:**

• Set the initial rate of CBR traffic to 0.5 Mb. Run the simulation, and plot the "Bytes Received" by node #s 4 and 5 (sinks for TCP and UDP traffic)
• Now, increment the rate up to 4 Mb, the link bandwidth, in steps of 0.5 Mb. Run the simulation and plot the graphs again.

How does the graphs change after each run? In particular, what's the nature of the graphs when the rate of CBR traffic is 50% of the bandwidth?

**Part 2: Behaviour of UDP**

• Reduce the bandwidth of the link 2-3 to say, 2 Mb. Repeat the above steps and observe the graphs in this case.

From the graphs plotted observe how UDP occupies a larger portion of the bandwidth. How does the behaviour change for other variations of TCP (Newreno, Vegas)?

**7. Write a TCL script to simulate the following scenario with ns2 simulator.**

Consider six nodes, (as shown in the figure below) moving within a flat topology of 700m x 700m.

The initial positions of nodes are 0 (150,300), 1 (300,500), 2 (500,500), 3 (300,100), 4 (500,100) and 5 (650,300) respectively. A TCP connection is initiated between node 0 (source) and node 5 (destination) through node 3 and node 4 i.e the route is 0-3-4-5.

At time $t = 3$ seconds the FTP application runs over it.

After time $t = 4.0$ sec, node 3 (300,100) moves towards node 1 (300,500) with a speed of 5.0m/sec and after some time the path break, then the data transmit with a new path via node 1 and node 2 i.e the new route 0-1-2-5.

The simulation lasts for 60 secs. In the above said case both the route has equal cost.

Use DSR as the routing protocol and the IEEE 802.11 MAC protocol. Now Analyze the trace file and determine when the use of second route commence, and Plot the number of packets received by each node over the entire time duration of the simulation.

**8. Simulate a wired network and demonstrate Distance Vector Routing algorithm.**
9. Simulate a network which will create congestion in the network. With the trace file created identify the points at which congestion occurs by writing sed / awk scripts. Also write a mechanism to correct/control the congestion.

Note: Network Free and open source software simulators like NS2 / NS3 could be used. If NS2 is used tcl scripting should be introduced. If NS3 is used c++ with python has to be introduced during first two or three weeks of the labs. Only above 9 experiments should be included for the laboratory exam.

**Note:** *In the examination each student should do one question out of the above 9 questions*
**SEMESTER IV**

**ANALYSIS AND DESIGN OF ALGORITHMS**

<table>
<thead>
<tr>
<th>Sub Code</th>
<th>IA Marks</th>
<th>Hrs/Week</th>
<th>Exam Hours</th>
<th>Total Hours</th>
<th>Exam Marks</th>
</tr>
</thead>
<tbody>
<tr>
<td>13MCA41</td>
<td>50</td>
<td>04</td>
<td>03</td>
<td>52</td>
<td>100</td>
</tr>
</tbody>
</table>

- **Introduction Fundamentals of the Analysis of Algorithm Efficiency, Brute Force** 12 Hours

- **Divide-and-Conquer** 8 Hours
  - Mergesort, Quicksort, Binary Search, Binary tree Traversals and related properties, Multiplication of large integers, Stressen’s Matrix Multiplication

- **Decrease-and-Conquer** 5 Hours
  - Insertion Sort, Depth First and Breadth First Search, Topological sorting, Algorithms for Generating Combinatorial Objects

- **Space and Time Tradeoffs** 6 Hours

- **Dynamic Programming** 5 Hours
  - Computing a binomial coefficient, Warshall’s and Floyd’s Algorithms, The Knapsack Problem and Memory Functions

- **Greedy Technique** 4 Hours
  - Prim’s Algorithm, Kruskal’s Algorithm, Dijkstra’s Algorithm, Knapsack

- **Limitations of Algorithm Power** 12 Hours
  - Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems

**Text Books:**

**Reference Books:**
3. Michael T Goodrich and Roberto Tamassia : Algorithm Design, Wiley India
Advanced JAVA Programming

Servlets: 8 Hours

JSP: 12 Hours
Overview of JSP Technology, Need of JSP, Benefits of JSP, Advantages of JSP, Basic syntax, Invoking java code with JSP scripting elements, creating Template Text, Invoking java code from JSP, Limiting java code in JSP, using jsp expressions, comparing servlets and jsp, writing scriptlets. For example Using Scriptlets to make parts of jsp conditional, using declarations, declaration example. Controlling the Structure of generated servlets: the JSP page directive, import attribute, session attribute, isElignore attribute, buffer and auto flush attributes, info attribute, errorPage and is errorPage attributes, is Thread safe Attribute, extends attribute, language attribute, Including files and applets in jsp Pages, using java beans components in JSP documents

Java Beans & Annotations: 6 Hours
Creating Packages, Interfaces, JAR files and Annotations. The core java API package, New java. Lang Sub package, Built-in Annotations. Working with Java Beans. Introspection, Customizers, creating java bean, manifest file, Bean Jar file, new bean, adding controls, Bean properties, Simple properties, Design Pattern events, creating bound properties, Bean Methods, Bean an Icon, Bean info class, Persistence Java Beans API.

JDBC: 8 Hours
Talking to Database, Immediate Solutions, Essential JDBC program, using prepared Statement Object, Interactive SQL tool. JDBC in Action Result sets, Batch updates, Mapping, Basic JDBC data types, Advanced JDBC data types, immediate solutions.

Introduction to EJB: 9 Hours
Server Side Component Models: 9 Hours
The Stateless Session Bean, the Stateful Session Bean, the Singleton Session Bean, Message-Driven Beans. EJB and PERSISTENCE. Persistence Entity manager Mapping Persistence objects, Entity Relationships.

Text Books:
2. Java 6 Programming Black Book, Dreamtech Press. 2012 (Chapter 17,18,19,20,21,22,27,28,29,30).
3. Andrew LeeRubinger, Bill Burke. Developing Enterprise Java Components. Enterprise JavaBeans 3.1.O’reilly. (Chapter 1,2,3,4,5,6,7,8,9,10,11).

Reference Books:
Advanced Web Programming

Subject Code : 13MCA43     I.A. Marks : 50
Hours/Week : 4     Exam Hours : 3
Total Hours : 52     Exam Marks : 100

1. Programming in Perl   7 Hours
Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays, Hashes, References, Functions, Pattern matching, File input and output; Examples.

2. CGI Scripting   6 Hours
What is CGI? Developing CGI Applications, Processing CGI, Introduction to CGI.pm, CGI.pm methods, Creating HTML Pages Dynamically, Using CGI.pm – An Example, Adding Robustness, Carp, Cookies

3. Building Web Applications with Perl   5 Hours
Uploading files, Tracking users with Hidden Data, Using Relational Databases, using libwww.

4. Introduction to PHP   8 Hours
Origins and uses of PHP, Overview of PHP, General syntactic characteristics, Primitives, operations and expressions, Output, Control statements, Arrays, Functions, Pattern matching, Form handling, Files

5. Building Web applications with PHP   6 Hours
Tracking users, cookies, sessions, Using Databases, Handling XML.

6. Introduction to Ruby   8 Hours
Origins and uses of Ruby, Scalar types and their operations, Simple input and output, Control statements, Arrays, Hashes, Methods, Classes, Code blocks and iterators, Pattern matching.

7. Introduction to Rails   4 Hours
Overview of Rails, Document requests, Processing forms, Rails applications with Databases, Layouts.

8. Introduction web 2.0,   4 Hours
What is Web 2.0?, Folksonomies and Web 2.0, Software As a Service (SaaS), Data and Web 2.0, Convergence, Iterative development, Rich User experience, Multiple Delivery Channels, Social Networking.

9. Web Services   4 Hours
Web Services: SOAP, RPC Style SOAP, Document style SOAP, WSDL, REST services, JSON format, What is JSON?, Array literals, Object literals, Mixing literals, JSON 0053yntax, JSON Encoding and Decoding, JSON versus XML.
Text Books:
3. Francis Shanahan: Mashups, Wiley India 2007(Chapters 1, 6)

Reference Books:
Advanced Computer Networks

Subject Code: 13MCA441    Internal Marks : 50
Hours / Week: 4            Exam Hours: 3
Total Hours: 52            External Marks: 100

1 Introduction  2 Hours
History of TCP/IP, TCP Applications and Services, Performance Study of TCP/IP, Meaning of TCP Performance?

2 TCP/IP Fundamentals  4 Hours
TCP, TCP Services, Header Format, Encapsulation in IP, Acknowledgment Mechanism, Retransmission Mechanism, Connection Establishment and Termination, Control and Sliding Window, Congestion Control, UDP, UDP Services, Header Format, Encapsulation in IP, IP Services, Fragmentation and Reassembly, Header Format and IP Version 6

3 Performance Measurement of TCP/IP Networks  3 Hours

4 TCP/IP Network Simulation  4 Hours
The Role of Simulation, Steps of a Systematic Simulation Study, Types of Simulations, Continuous versus Discrete Event, Terminating versus Steady State, Synthetic versus Trace-Driven Simulation, Simulation Validation and Verification, Confidence Level of Simulation Results, Confidence Level Formula, Terminating Simulation, Steady-State Simulation, Common Simulation Mistakes, Simulation with Self-Similar Traffic. Network Simulators: Model Construction and Parameter Setting Data Collection, Simulation Execution, Presentation of Results and Examples of TCP/IP Simulation

5 TCP Modeling  5 Hours

6. TCP/IP Performance over Wireless Networks  3 Hours

7. TCP/IP Performance over Mobile Networks  3 Hours
Cellular and Ad Hoc Networks: TCP Performance in Cellular Networks, Mobile IP, Impact of Mobility on TCP Performance, Approaches to Improve TCP Performance. TCP Performance in Ad Hoc Networks, Dynamic Source Routing, Impact of Mobility on TCP Performance,
Approaches to Improve TCP Performance

8. TCP/IP Performance over Optical Networks 3 Hours

9. TCP/IP Performance over Satellite Networks 8 Hours

10. TCP/IP Performance over Asymmetric Networks 3 Hours

11. New TCP Standards and Flavors 3 Hours
Duplicate Acknowledgments and Fast Retransmit, Fast Recovery and TCP Reno, TCP NewReno, TCP with Selective Acknowledgments, Forward Acknowledgments, TCP Vegas, Overview of Other Features and Options and Performance Comparison of TCP Flavors

12. Active Queue Management in TCP/IP Networks 6 Hours

13. TCP Implementation 5 Hours
Text Book:

Chapters 1,2,3,4,5,6,7,8,9,10,11,12,13 (excluding those topics which are not in the syllabus)

Reference Books:

TCP/IP Illustrated (Volume I, Volume II and Volume III), W. Richard Stevens, *Addison-Wesley*
Data Warehousing and Data Mining

Data Warehousing and OLAP 8 Hours
Data Warehouse basic concepts, Data Warehouse Modeling, Data Cube and OLAP

Data Mining 6 Hours
Introduction, What is Data Mining, Motivating Challenges, Data Mining Tasks, Which technologies are used, which kinds of applications are targeted by Data Mining

Data Mining 6 Hours
Types of Data, Data Mining Applications, Data Preprocessing

Association Analysis: Basic Concepts and Algorithms 8 Hours
Frequent Item set Generation, Rule Generation, Compact Representation of Frequent Item sets, Alternative methods for generating Frequent Item sets, FP Growth Algorithm, Evaluation of Association Patterns

Classification 12 Hours

Clustering Techniques 8 Hours
Overview, Features of cluster analysis, Types of Data and Computing Distance, Types of Cluster Analysis Methods, Partitional Methods, Hierarchical Methods, Density Based Methods, Quality and Validity of Cluster Analysis

Outlier Analysis 4 Hours
Outlier detection methods, Statistical Approaches, Clustering based applications, Classification based approached

Text Books:

Reference Books:
2. Jiawei Han and Micheline Kamber: Data Mining - Concepts and Techniques, 2nd Edition, Morgan Kaufmann Publisher, 2006.
Mobile Computing and Wireless Communications

Subject Code: 13MCA443  IA Marks : 50
Hours/Week: 04  Exam Hours : 03
Total Hours: 52  Exam Marks : 100

Mobile Computing Architecture:  6 Hours

Wireless Networks – 1: GSM and SMS  7 Hours
Global Systems for Mobile Communication (GSM and Short Service Messages (SMS): GSM Architecture, Entities, Call routing in GSM, PLMN Interface, GSM Addresses and Identities, Network Aspects in GSM, Mobility Management, GSM Frequency allocation. Introduction to SMS, SMS Architecture, SM MT, SM MO, SMS as Information bearer, applications

Wireless Networks – 2: GPRS  6 Hours
GPRS and Packet Data Network, GPRS Network Architecture, GPRS Network Operations, Data Services in GPRS, Applications for GPRS, Billing and Charging in GPRS

Wireless Networks – 3: CDMA, 3G and WiMAX  7 Hours
Spread Spectrum technology, IS-95, CDMA versus GSM, Wireless Data, Third Generation Networks, Applications on 3G, Introduction to WiMAX.

Mobile Client  6 Hours
Moving beyond desktop, Mobile handset overview, Mobile phones and their features, PDA, Design Constraints in applications for handheld devices. Mobile IP: Introduction, discovery, Registration, Tunneling, Cellular IP, Mobile IP with IPv6

Mobile OS and Computing Environment  7 Hours

Building, Mobile Internet Applications  6 Hours
Thin client: Architecture, the client, Middleware, messaging Servers, Processing a Wireless request, Wireless Applications Protocol (WAP) Overview, Wireless Languages: Markup Languages, HDML, WML, HTML, cHTML, XHTML, VoiceXML.

J2ME  7 Hours
Introduction, CDC, CLDC, MIDP; Programming for CLDC, MIDlet model, Provisioning, MIDlet life-cycle, Creating new application, MIDlet event handling, GUI in MIDP, Low level GUI Components, Multimedia APIs; Communication in MIDP, Security Considerations in MIDP.
Text Books:

Reference Books:
Software Testing and Practices

Basics of Software Testing  7 hours
Humans, Errors and Testing, Software Quality; Requirements, Behavior and Correctness, Correctness Vs Reliability; Testing and Debugging; Test Metrics; Software and Hardware Testing; Testing and Verification; Defect Management; Execution History; Test Generation Strategies; Static Testing; Test Generation from Predicates.

Basic Principles, Test case selection and Adequacy  6 hours
Sensitivity, Redundancy, Restriction, Partition, Visibility and Feedback, Test Specification and cases, Adequacy Criteria, Comparing Criteria

A perspective on Testing, Examples  7 Hours
Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Level of testing, Examples: Generalized pseudocode, The triangle problem, the NextDate function, The commission problem, The SATM (Simple Automation Teller Machine) problem, The currency converter, Saturn windshield wiper.

Boundary value testing, Equivalence class testing, Decision table based testing  7 Hours
Boundary value analysis, Robustness testing, Worst-case testing, special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for triangle problem, NextDate function and commission problem, Guidelines and observations, Decision tables, Test cases for triangle problem.

Path Testing, Data flow testing  7 Hours
DD Paths, Test coverage metrics, Basis path testing, guidelines and observations, Definition Use testing, Slice based testing, Guidelines and observations.

Levels of Testing, Integration Testing  6 Hours
Traditional view of testing levels, Alternative life cycle models, the SATM systems, separating integration and system testing, Guidelines and observations.

Fault Based Testing  6 Hours
Assumptions in fault-based testing, Mutation Analysis, Fault-based Adequacy Criteria; Variations on mutation Analysis; From Test case specification to Test Cases, Scaffolding, Generic vs specific Scaffolding, Test Oracles, Self checks as oracles, Capture and Replay.

Planning and Monitoring the Process, Documenting Analysis and Test  6 Hours
Quality and Process, Test and Analysis strategies and plans, Risk Planning, Monitoring the Process, Improving the process, The quality team, Organizing documents, Test strategy document, Analysis and test plan, Test design specifications documents, Test and analysis reports.
Text Books
2. MauroPezze, Michael Young, Software testing and Analysis- Process, Principles and Techniques, Wiley India, 2012

Reference Books
Theory of Computation (Finite Automata and Formal Languages)

Subject Code: 13MCA445
IA Marks : 50
Hours/Week: 04
Exam Hours : 03
Total Hours : 52
Exam Marks: 100

Introduction and Finite Automata:

10 Hours
What is (not) a computer, The idea of computing, Computing Machines and Languages, What is the Science of Computing, Programming, Data Structures, Algorithms and Science, Birth of Science computing, Computability, Undecidability, Intractability and Intelligence, Why Study Science computing and Key Ideas, Automata- The idea of computing Machine, Automata Definition, Constructing Simple Automata, Handling End Condition, Handling Reject States, A Step-by-Step model for constructing Automata, States as Memory, Why Finite number of states, Constructing more complex Automata, Mantras for constructing Automata, Limitations of Finite Automata, Automata with Combinatorial States

NFA and Regular Expression

7 Hours
The idea of Non-Determinism, Constructing Non-Deterministic Automata, Eliminating Non-Deterministic: converting NFA to DFA, Jumping States without Input, A method for minimizing Automata, Finite State Transducers, The idea of formal languages, Languages of Automata, Regular Expression, Constructing Regular Expressions, Converting Regular Expressions to Automata, Equivalence of Regular Expressions, Method for Constructing Regular Expressions, Regular Expressions in Practice

Regular Grammars and Languages

7 Hours
The idea of Grammar, The ideas of parsing and Derivation, Grammars for Regular Languages, Constructing Regular Grammars, converting automata to regular grammars, converting regular grammars to automata, constructing regular grammars: mantras, Closure properties, Answering questions about regular languages, Why are some languages not regular, The Pigeonhole Principle and Pumping Lemma, Using Pumping Lemma an Adversarial Game.

Context Free Grammars

7 Hours
The idea and nature of context free grammar, Constructing Context free grammars (LGs and Non LGs), Introduction to Parsing, Ambiguity and Eliminating ambiguity, The idea of Chomsky normal form, Converting to Chomsky normal form, The ideas of Greibach Normal form, Simple Linear and other grammars.

Pushdown Automata and Nature of Context Free Languages

7 Hours
Machines for Context Free Languages, Adding Memory: Why Stack Behavior, Constructing PDAs, Constructing CFGs to PDAs, Converting PDAs to CFGs, Non-determinism in PDAs, The CFL-CFG-PDA Triad, Closure Properties, Union of CFLs, Answering Questions about CFLs, Why are some languages not context-free, The pumping lemma for context free languages.

Turing Machines

8 Hours
The Chomsky Hierarchy
Languages, Grammars and Machines, Recursively Enumerable Languages, Counting Alphabets, Languages and Computing Machines, The idea of Enumeration, The idea of Diagonalization, The ideas of Acceptance and Membership, Recursive Languages, Context Sensitive Languages and Grammars, The ideas of context, Other Grammars and Automata, Linear and Deterministic Context-Free Languages.

Text Books:

Reference Books:
2. Introduction to Automata Theory, Languages, and Computation, Addison Wesley Publishing company, 2010
**Digital Image processing**

<table>
<thead>
<tr>
<th>Subject Code : 13MCA446</th>
<th>IA Marks : 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hrs/Week : 04</td>
<td>Exam Hours: 03</td>
</tr>
<tr>
<td>Total Hrs. : 52</td>
<td>Exam Marks: 100</td>
</tr>
</tbody>
</table>

**UNIT 1: Introduction**  
10 Hours  

**UNIT 2: Image Enhancement in the Spatial Domain**  
8 hours  

**UNIT 2: Image Enhancement in the Frequency Domain**  
8 hours  

**UNIT 4: Morphological Image Processing and Image Segmentation**  
10 Hours  
Dilation and erosion, opening and closing, Hit-or-Miss transformations, basic morphological algorithms, Detection of discontinues, edge linking and boundary detection, thresholding, region –based segmentation.

**UNIT 5: Representation and Descriptors**  
8 Hours  

**UNIT 6: Use of Image Processing in Pattern Recognition**  
8 Hours  
Introduction to the tools of Matlab and Open CV. Case study on Object Identification, Biometrics and Content Based Image retrieval.

**Text Books:**

**Reference Books:**
2. Vipul Singh, Digital Image Processing With Matlab & LabView, Reed Elsevier India Pvt Ltd, 2013,
Introduction

Classical Encryption Technique
Symmetric Cipher Model, Substitution Techniques, Transposition Techniques.

Block Ciphers, Data Encryption Standard and Advanced Encryption Standard

Public Key Cryptography and Key Management
Principles of Public Key Cryptosystem, RSA algorithm, Key management, Diffie Hellman Key exchange

Message Authentication and Hash Function
Authentication Requirement, Authentication Functions, Message Authentication Code, Hash Functions, Digital Signatures, Digital Signature Standard

Authentication Applications
Kerberos, X.509 Authentication Service

Electronic Mail Security
Pretty Good Privacy (PGP), S/MIME

IP Security
IP Security Overview; IP Security Architecture; Authentication Header; Encapsulating Security Payload; Combining Security Associations; Key Management.

Web Security
Web security Considerations; Secure Socket layer (SSL) and Transport layer Security (TLS); Secure Electronic Transaction (SET)

System Security
Intruders, Intrusion Detection, Firewall Design Principles- Characteristics, Types of Firewall and Firewall Configuration.
**Text Books:**


**Reference Book:**

Network Management

Subject code: 13MCA452  
IA Marks: 50  
Hours/ Week: 04  
Exam Hours: 3  
Total Hours : 52  
Exam Marks: 100

Requirements for the Management of Networked Systems  05 Hours  
Management Scenarios, Management functions, Organizational aspects of Management,  
Time aspects of Management

IP Network Management  07 Hours  
Choosing to manage the network, Choosing a configuration method, Management  
information Base, Simple Network Management Protocol, Extensible markup Language,  
Common Object Request Broker Architecture.

IP-Based Service Implementation and Network Management  08 Hours  
Simple Network Management Protocol, Ip- Based Service Implementation-OSS,  
Provisioning Issues, Network Management Issues, OSS Architecture

Network Management Architecture  06 Hours  
Background, Defining Network Management, Network Management Mechanisms,  
Architectural Considerations.

SLA and Network Monitoring  05 Hours  
Passive and Active Network Monitoring, Passive Network Monitoring,  
Active Network Monitoring.

MPLS Network Management: AN Introduction  05 Hours  
A brief Introduction to MPLS, MPLS Applications, Key Aspects of MPLS  
Network Management, Management Information Base Modules for MPLS.

MPLS Management Interfaces  05 Hours  
The basics of Management Interfaces, Command line interface, CORBA, XML, Bulk  
File Transfer, Simple Network Management Protocol

Optical Networks: Control and Management  05 Hours  
Network Management functions, Optical Layer Services and Interfacing, Layers within  
the Optical Layer, Multivendor Interoperability, Performance and Fault Management,  
Configuration

Web-Based Management  06 Hours  
NMS with Web Interface and Web- Based Management Web Interface to SNMP  
Management, Embedded Web- Based Management, Desktop management interface, Web- 
Based Enterprise Management, WBEM: Windows Management Instrumentation, Java
management Extensions, Management of a Storage Area Network: Future Directions.

Text Books:
1. Network Management- Know it all by Adrian Farrel, Elsevier publications. Chapter 1-8
2. Network Management- Principles and Practice, Mani Subramaniam, Pearson Education. Chapter 14

Reference Books:
1. Network Management, Morris, Pearson Education
2. Practical Guide to SNMPv3 and Network Management, David Zeltserman, PHI.
NOSQL

Subject Code::  13MCA453  
I.A. Marks  :  50

Hours/Week   03  
Exam Hours  :  3

Total Hours :  52  
Exam Marks  :  100

Introduction to NOSQL  
6 Hours  
Definition of NOSQL, History of NOSQL and Different NOSQL products, Exploring MondoDB Java/Ruby/Python, Interfacing and Interacting with NOSQL

NOSQL Basics  
12 Hours  
NOSQL Storage Architecture, CRUD operations with MongoDB, Querying, Modifying and Managing NOSQL Data stores, Indexing and ordering datasets (MongoDB/CouchDB/Cassandra)

Advanced NOSQL  
8 Hours  
NOSQL in CLOUD, Parallel Processing with Map Reduce, BigData with Hive

Working with NOSQL  
10 Hours  
Surveying Database Internals, Migrating from RDBMS to NOSQL, Web Frameworks and NOSQL, using MySQL as a NOSQL

Developing Web Application with NOSQL and NOSQL Administration  
16 Hours  
Php and MongoDB, Python and MongoDB, Creating Blog Application with PHP, NOSQL Database Administration

Text Books

1. “Professional NOSQL” by Shashank Tiwari, 2011, WROX Press (Chapter 1,2,3,4,5,6,7, 8, 9,10.11.12.13.15)
2. The Definitive guide to MongoDB, The NoSQL Database for Cloud and Desktop Computing, Apress 2010 (Chapter 6,7,8,9)
# Software Architectures

<table>
<thead>
<tr>
<th>Introduction</th>
<th>06 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>What software architecture is and what it is not; Architectural Structures and views; Architectural patterns; What makes a “good” architecture? Why is software important?</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Context of Software Architecture</th>
<th>04 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical Context; Project life-cycle context; Business context; Professional context; Stake holders; How is Architecture influenced? What Do Architecture influence?</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Understanding Quality Attributes</th>
<th>12 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture &amp; Requirements; Functionality; quality attribute considerations; Specifying and achieving Quality attribute requirements; Guiding quality design decisions; Availability; Interoperability; Modifiability; Performance; Security; Testability; Usability</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quality Attribute modeling and Analysis</th>
<th>06 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Modeling Architecture to enable quality attribute analysis; Quality attribute check lists; Through experiments and Back-of-the envelope analysis; Experiments; Simulations and prototypes; Analysis at different stages of the life cycle</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Architecture and requirements</th>
<th>06 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gathering ASRs from requirements documents; ASRs by interviewing stake holders; ASRs by understanding the business; capturing ASRs in a utility tree; Typing the methods together</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Designing an Architecture</th>
<th>03 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design strategy; the attribute driven design methods; the steps of ADD</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Documenting Software Architecture</th>
<th>06 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses and Audiences for architecture documentation; Notations, View and Behavior; Documentation and quality attributes</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Architecture, Implementation &amp; Testing</th>
<th>03 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture and implementation; Architecture and testing</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Architectural Patterns</th>
<th>06 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction to patterns; From Mud to structure; Layers; Pipes and filters; Blackboard; Distributed systems; Broker; Interactive systems; Model-view-control; Presentation-abstraction-control; Adaptable systems; Microkernel</td>
<td></td>
</tr>
</tbody>
</table>
Text Books:
1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 3d Edition, Pearson Education, 2013 (Listed Topics only from Chapters 1,2,3,4,5,6,7,8,9,10,11,14,16,17,18,19)
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal: Pattern- Oriented Software Architecture, A System of Patterns, Volume 1, John Wiley and Sons, 2012 (chapter 2)

Reference Books:
Enterprise Resource Planning

Subject Code : 13MCA455 I.A. Marks : 50
Hours/Week : 4 Exam Hours : 3
Total Hours : 52 Exam Marks : 100

Unit I Introduction To ERP 9 Hours
Overview, Benefits of ERP, ERP and Related Technologies, Business Process Reengineering, Data Warehousing, Data Mining, On–line Analytical Processing, Supply Chain Management.

Unit II ERP Implementation 12 Hours

Unit III Business Modules 10 Hours

Unit IV ERP Market 10 Hours

Unit V ERP – Present And Future 11 Hours
Turbo Charge the ERP System, EIA, ERP and E-Commerce, ERP and Internet, Future Directions in ERP.

TextBooks

Reference Books
Mobile Applications

Subject Code: 13MCA456  I.A. Marks : 50
Hours/Week : 04  Exam Hours: 03
Total Hours : 52  Exam Marks: 100

Foundation  6 Hours
Mobile devices vs. desktop devices; ARM and intel architectures, Power Management, Screen resolution, Touch interfaces, Application deployment (App Store, Google Play, Windows Store); Native vs. web applications

Frameworks and Tools  10 Hours
Development Environments (XCode, Eclipse, VS2012, PhoneGAP, etc.); Development Tools (HTML5, CSS, JavaScript, JQuery); Mobile-specific enhancements (Browser-detection, Touch interfaces, Geolocation, Screen orientation); Mobile browser “interpretations” (Chrome/Safari/Gecko/IE).

Mobile OS Architectures  8 Hours
Mobile OS Architectures (Android, iOS, Windows); Mobile OS (Darwin, Linux, Windows); Runtime Environments (Objective-C, Dalvik, winRT), Mobile Agents and Peer-to-Peer Architecture

Performance Management  6 Hours
Memory Management, Power Management, Security, Synchronization and Replication of Mobile Data, Getting the Model right, Storing and Retrieving Data

Developing an Application  8 Hours
Building a simple “Hello World” App (Android, iOS, Windows); App-structure, built-in Controls, file access, basic graphics; Building useful apps; Database, Network, File access; Packaging and Deployment

System-level Apps  6 Hours
Native programming (Android), Low-level programming (iOS), Low-level APIs (Windows).

Advanced Topics: Power Management, Augmented Reality, Mobile Device Security  8 Hrs
Wake locks and assertions, Low-level OS support, Writing power-smart applications, GPS, Accelerometer, Camera, Mobile malware, Device protections, Rooting (Android), Jailbreaking (iOS), Defenestration (Windows)

Text Books:
1 Jeff McWherter, Scott Gowell: Professional Mobile Application Development, John Wiley & Sons, Aug 2012
3 HTML5, CSS3 and Jquery with Adobe Dreamweaver CS 5.5 Learn by Video, David Powers, Richard, Trade paperback, Peachpit Press, 2011
References:

4. Reto Meier: Professional Android 4 Application Development, Wiley-India
Implement the following using C/C++ Language.

1. Implement Recursive Binary search and Linear search and determine the time required to search an element. Repeat the experiment for different values of n, the number of elements in the list to be searched and plot a graph of the time taken versus n.

2. Sort a given set of elements using the Heapsort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.

3. Sort a given set of elements using Merge sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.

4. Obtain the Topological ordering of vertices in a given igraph.

5. Implement 0/1 Knapsack problem using dynamic programming.

6. From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.

7. Sort a given set of elements using Quick sort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n.

8. Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.

9. Print all the nodes reachable from a given starting node in a digraph using BFS method.

10. Check whether a given graph is connected or not using DFS method.

11. Find a subset of a given set $S = \{s_1, s_2, \ldots, s_n\}$ of n positive integers whose sum is equal to a given positive integer d. For example, if $S = \{1, 2, 5, 6, 8\}$ and $d = 9$ there are two solutions $\{1, 2, 6\}$ and $\{1, 8\}$. A suitable message is to be displayed if the given problem instance doesn't have a solution.

    b. Find the Binomial Co-efficient using Dynamic Programming.

13. Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.
14.  a. Implement Floyd’s algorithm for the All-Pairs-Shortest-Paths Problem.
    b. Compute the transitive closure of a given directed graph using Warshall’s algorithm.

15. Implement N Queen’s problem using Back Tracking.

**Note: In the examination questions must be given based on above lots.**
1. Write a JAVA Servlet Program to implement a dynamic HTML using Servlet (user name and Password should be accepted using HTML and displayed using a Servlet).

2. Write a JAVA Servlet Program to Auto Web Page Refresh (Consider a webpage which is displaying Date and time or stock market status. For all such type of pages, you would need to refresh your web page regularly; Java Servlet makes this job easy by providing refresh automatically after a given interval).

3. Write a JAVA Servlet Program to implement and demonstrate get() and Post methods(Using HTTP Servlet Class).

4. Write a JAVA Servlet Program using cookies to remember user preferences.

5. a. Write a JAVA JSP Program to implement verification of a particular user login and display a Welcome page.
   b. Write a JSP program to demonstrate the import attribute.

6. Write a JAVA JSP Program which uses jsp:include and jsp:forward action to display a Webpage.

7. Write a JAVA JSP Program which uses <jsp:plugin> tag to run a applet.

8. Write a JAVA JSP Program to get student information through a HTML and create a JAVA Bean class, populate Bean and display the same information through another JSP.

9. Write a JAVA Program to insert data into Student DATA BASE and retrieve info based on particular queries(For example update, delete, search etc…).

10. Write a JSP program to implement all the attributes of page directive tag.

11. An EJB application that demonstrates Session Bean (with appropriate business logic).

12. An EJB application that demonstrates MDB (with appropriate business logic).

13. An EJB application that demonstrates persistence (with appropriate business logic).

*Note: In the examination each student should do one question out of the above 13 questions*
Mini Project-I Guidelines

Subject Code : 13MCA48
IA Marks : 50
Hours/week : 3
Exam Hours : 03
Total Hours : 42
Exam Marks : 50

PART A

1. Write a perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.
2. Create XHTML form with Name, address line1, address line2 and email text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on name.
3. Write a PHP program to read student data from an XML file and store into the MySQL database. Retrieve and display using SEARCH function.
4. Build a Rails application to accept book information viz. accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

PART B

Develop a web application project using the languages and concepts learnt in the theory and exercises listed in part A with a good look and feel effects. You can use any web technologies and frameworks and databases.

Note:
1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report (25-30 pages) that must include the following
   a. Introduction
   b. Requirement Analysis
   c. Software Requirement Specification
   d. Analysis and Design
   e. Implementation
   f. Testing
4. The report must be evaluated for 10 Marks. Demonstration and Viva for 20 Marks.

Instructions:
1. In the examination, one exercise from part A is to be asked for 20 marks.
2. Mini project student group size is limited to two students only.
3. The mini project under part B has to be evaluated for 30 marks.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.
SEMESTER V
Object-Oriented Modeling and Design Patterns

Subject Code: 13MCA51
I.A. Marks : 50
Hours/Week: 4
Exam Hours: 3
Total Hours: 52
Exam Marks: 100

1. Introduction, Modeling Concepts
What is Object Orientation? What is OO development? OO themes; Evidence for usefulness of OO development; OO modeling history. Modeling as Design Technique: Modeling; abstraction; The three models.

2. Class Modeling and Advanced Class Modeling:
Object and class concepts; Link and associations concepts; Generalization and inheritance; A sample class model; Navigation of class models; Practical tips. Advanced object and class concepts; Association ends; N-array associations; Aggregation; Abstract classes; Multiple inheritance; Metadata; Reification; Constraints; Derived data; Packages; Practical tips

3. State Modeling and Advanced State Modeling
State Modeling: Events, States, Transitions and Conditions; State diagrams; State diagram behavior; Practical tips. Advanced State Modeling: Nested state diagrams; Nested states; Signal generalization; Concurrency; A sample state model; Relation of class and state models; Practical tips.

4. Interaction Modeling and Advanced Interaction Modeling
Interaction Modeling: Use case models; Sequence models; Activity models. Use case relationships; Procedural sequence models; Special constructs for activity models.

5. Process Overview, System Conception
Process Overview: Development stages; Development life cycle. System Conception: Devising a system concept; elaborating a concept; preparing a problem statement.

6. Domain Analysis and Application Analysis
Overview of analysis; Domain class model; Domain state model; Domain interaction model; Iterating the analysis. Application Analysis: Application interaction model; Application class model; Application state model; adding operations.

7. System Design and Class Design
Overview of system design; Estimating performance; Making a reuse plan; Breaking a system into sub-systems; Identifying concurrency; Allocation of sub-systems; Management of data storage; Handling global resources; Choosing a software control strategy; Handling boundary conditions; Setting the trade-off priorities; Common architectural styles; Architecture of the ATM system as the example. Class Design: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recursing downwards, Refactoring; Design optimization; Reification of behavior; Adjustment of inheritance; Organizing a class design; ATM example.

8. Patterns
What is a pattern and what makes a pattern? Pattern categories; Relationships between
patterns; Pattern description.

9. Design Patterns  10 Hrs
Introduction, structural decomposition, Organization of work, Model View Controller;
Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber;
Management Patterns: Command processor; Whole Part, Master Slave, View Handler;

Text Books:
1. Michael Blaha, James Rumbaugh: Object-Oriented Modeling and Design with UML,
2. Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal:
   Pattern-Oriented Software Architecture, A System of Patterns, Volume 1, John
   Wiley and Sons, 2006. (Chapters 1, 3)

Reference Books:
5. E. Gamma, R. Helm, R. Johnson, J. Vlissides: Design Patterns-Elements of Reusable
   Object- Oriented Software, Addison-Wesley, 1995.
6. Simon Bennett, Steve McRobb and Ray Farmer: Object-Oriented Systems Analysis and
7. Rumbaugh, Blaha, Premerhani, Eddy, Lorensen; Object Oriented Modeling and Design, PHI
   Latest Edition
System Simulation and Modeling

Subject Code: 13MCA52
IA Marks : 50
Hours/Week: 04
Exam Hours : 03
Total Hours: 52
Exam Marks : 100

Introduction
When simulation is the appropriate tool and when it is not appropriate; Advantages and disadvantages of Simulation; Areas of application; Systems and system environment; Components of a system; Discrete and continuous systems; Model of a system; Types of Models; Discrete-Event System Simulation; Steps in a Simulation Study.

Statistical Models in Simulation
Review of terminology and concepts; Random Variables, Probability Distribution, Probability distribution function, Useful statistical models; Discrete distributions; Continuous distributions; Poisson process; Empirical distributions.

Random-Number Generation, Random-Variate Generation
Properties of random numbers; Generation of pseudo-random numbers; Techniques for generating random numbers; Tests for Random Numbers, Random-Variate Generation: Inverse transform technique; Acceptance-Rejection technique; Special properties.

Queuing Models
Characteristics of queuing systems; Queuing notation Simulation Examples: Queuing, Inventory System

General Principles, Simulation Software

Input Modeling
Data Collection; Identifying the distribution with data; Parameter estimation; Goodness of Fit Tests; Fitting a non-stationary Poisson process; Selecting input models without data; Multivariate and Time-Series input models, uniformity and independence, Chi-Square test, K-S Test.

Verification and Validation
Verification, Calibration, and Validation; Optimization: Model building, verification and validation; Verification of simulation models; Calibration and validation of models.

Estimation of Absolute Performance & Computer System Simulation
Types of simulations with respect to output analysis; Stochastic nature of output data; Absolute measures of performance and their estimation; Output analysis for terminating simulations; Output analysis for steady-state simulations.
Text Books:

Reference Books:
3. Simulation 5ed Ross Elsevier
4. Theory of modeling and simulation, Zeiglar, Elsevier
Programming Using C#.NET

Subject: 13MCA53
IA Marks: 50
Hours/Week: 04
Exam Hours: 03
Total Hours: 52
Exam Marks: 100

Getting started with .NET Framework 4.0 04 Hours

Introducing C# 06 Hours
Need of C#, C# Pre-processor Directives, Creating a Simple C# Console Application, Identifiers and Keywords. Data Types, Variables and Constants: Value Types, Reference Types, Type Conversions, Boxing and Undoing, Variables and Constants. Expression and Operators: Operator Precedence, Using the ?? (Null Coalescing) Operator, Using the :: (Scope Resolution) Operator and Using the is and as Operators. Control Flow statements: Selection Statements, Iteration Statements and Jump Statements.

Namespaces, Classes, Objects and Structures 06 Hours

Object-Oriented Programming 05 Hours

Delegates and Events and Exception Handling 05 Hours
Graphical User Interface with Windows Forms  

Data Access with ADO.NET  

Web App Development with ASP.NET  

Case study: Database-Driven ASP.NET Guestbook, Building a Web Form that Displays Data from a Database, Modifying the Code-Behind File for the Guestbook Application, ASP.NET AJAX: Traditional Web Applications, Ajax Web Applications, Testing an ASP.NET Ajax application, the ASP.NET Ajax Control Toolkit. Case study: Password-Protected Books Database Application

Text Books:
1. .NET 4.0 Programming (6-in-1), Black Book, Kogent Learning Solutions Inc., Wiley- Dream Tech Press. (Chapters: 1,10,11,12,13,14 and 19).

References Books:
2. Bart De Smet: C# 4.0 Unleashed, Pearson Education- SAMS Series.
Mobile and Adhoc Sensor Networks

Subject Code: 13MCA541  
I.A. Marks : 50  
Hours/Week : 4  
Exam Hours: 3  
Total Hours : 52  
Exam Marks: 100

Unit I  
12 Hours  

Unit II  
10 Hours  

Unit III  
10 Hours  

Unit IV  
10 Hours  

Unit V Text  
10 Hours  

Text Book:  

References  
Parallel Computing

Subject Code: 13MCA542
Hours / Week: 4
Total Hours: 52
IA | Marks: 50
Exam Hours: 3
Exam Marks: 50

1. Introduction to Parallel Computing
   Need of Performance, Building Parallel Systems, Why to Write Parallel Programs? How to Write Parallel Programs? Approach: Concurrent, Parallel, Distributed

2 Parallel Hardware and Parallel Software
   Background, Modifications to the von Neumann Model, Parallel Hardware, Parallel Software, Input and Output, Performance, Parallel Program Design and Writing and Running Parallel Programs

3 Distributed Memory Programming with MPI
   Getting Started, The Trapezoidal Rule in MPI, Dealing with I/O, Collective Communication, MPI Derived Data types, A Parallel Sorting Algorithm

4 Shared Memory Programming with Pthreads
   Processes, Threads and Pthreads, Hello, World program, Matrix-Vector Multiplication, Critical Sections Busy-Waiting, Mutexes, Producer-Consumer Synchronization and Semaphores, Barriers and Condition Variables, Read-Write Locks, Caches, Cache-Coherence, and False Sharing and Thread-Safety

5 Shared Memory Programming with OpenMP

6 Parallel Program Development and Parallel Algorithms
   Two N-Body Solvers, Tree Search and Case Studies

Text Books:

Reference Books:
1. Using OpenMP: Portable Shared Memory Parallel Programming, Gabriele Jost and Ruud van der Pas The MIT Press (October 12, 2007)
Multimedia Systems

Subject Code: 13MCA543
Hours/Week : 4
Total Hours : 52
I.A. Marks : 50
Exam Hours : 3
Exam Marks : 100

Introduction, Media and Data Streams, Audio Technology 7 Hours
Multimedia Elements; Multimedia Applications; Multimedia Systems Architecture; Evolving Technologies for Multimedia Systems; Defining Objects for Multimedia Systems; Multimedia Data Interface Standards; The need for Data Compression; Multimedia Databases.
Sound: Frequency, Amplitude, Sound Perception and Psychoacoustics; Audio Representation on Computers; Three Dimensional Sound Projection; Music and MIDI Standards; Speech Signals; Speech Output; Speech Input; Speech Transmission.

Graphics and Images, Video Technology, Computer-Based Animation 7 Hours
Capturing Graphics and Images Computer Assisted Graphics and Image Processing; Reconstructing Images; Graphics and Image Output Options. Basics; Television Systems; Digitalization of Video Signals; Digital Television; Basic Concepts; Specification of Animations; Methods of Controlling Animation; Display of Animation; Transmission of Animation; Virtual Reality Modeling Language.

Data Compression 12 Hours
Storage Space; Coding Requirements; Source, Entropy, and Hybrid Coding; Basic Compression Techniques; JPEG: Image Preparation, Lossy Sequential DCT-based Mode, Expanded Lossy DCT-based Mode, Lossless Mode, Hierarchical Mode. H.261 (Px64) and H.263: Image Preparation, Coding Algorithms, Data Stream, H.263+ and H.263L; MPEG: Video Encoding, Audio Coding, Data Stream, MPEG-2, MPEG-4, MPEG-7; Fractal Compression.

Optical Storage Media 6 Hours
History of Optical Storage; Basic Technology; Video Discs and Other WORMs; Compact Disc Digital Audio; Compact Disc Read Only Memory; CD-ROM Extended Architecture; Further CD-ROM-Based Developments; Compact Disc Recordable; Compact Disc Magneto-Optical; Compact Disc Read/Write; Digital Versatile Disc.

Content Analysis 6 Hours
Simple Vs. Complex Features; Analysis of Individual Images; Analysis of Image Sequences; Audio Analysis; Applications.

Data and File Format Standards 7 Hours
Rich-Text Format; TIFF File Format; Resource Interchange File Format (RIFF); MIDI
Multimedia Application Design 7 Hours
Multimedia Application Classes; Types of Multimedia Systems; Virtual Reality Design; Components of Multimedia Systems; Organizing Multimedia Databases; Application Workflow Design Issues; Distributed Application Design Issues.

Text Books:

Reference Books:
UNIT – 1 8 Hours
INTRODUCTION: Machine perception, pattern recognition systems, design cycle, learning and adaptation, Applications of pattern recognition.

UNIT – 2 9 Hours
PROBABILITY: Introduction, probability of events, random variables, Joint distributions and densities, moments of random variables, estimation of parameters from samples, minimum risk estimators.

UNIT – 3 10 Hours
STATISTICAL DECISION MAKING: Introduction, Baye’s Theorem, multiple features, conditionally independent features, decision boundaries, unequal costs of error, estimation of error rates, the leavingone- out technique. Characteristic curves, estimating the composition of populations.

UNIT – 4 9 Hours
NONPARAMETRIC DECISION MAKING: Introduction, histograms, Kernel and window estimators, nearest neighbor classification techniques, adaptive decision boundaries, adaptive discriminate Functions, minimum squared error discriminate functions, choosing a decision making technique.

UNIT – 5 8 Hours
UNSUPERVISED LEARNING AND CLUSTERINGS: Unsupervised Bayesian learning, data decryption and clustering, criterion functions and clustering, Hierarchical clustering, Online clustering, component analysis.

UNIT – 6 8 Hours
ARTIFICIAL NEURAL NETWORKS: Introduction, nets without hidden layers, nets with hidden layers, the back Propagation algorithms, Hopfield nets, an application.

TEXT BOOKS:
2. Pattern Recognition and Image Analysis, Earl Gose, Richard J and Steve J, PHI
3. Pattern recognition (Statistical, structural and Neural Approaches), Robert Schalkoff

REFERENCE BOOKS
Service Oriented Architectures (SOA)

Introduction to SOA, Evolution of SOA  6 hours
Fundamentals of SOA, Common characteristics of contemporary SOA, Common tangible benefits of SOA. A SOA timeline (from XML to Web Services to SOA), The continuing evolution of SOA (standards organizations and Contributing vendors), The roots of SOA (comparing SOA to Past Architectures)

Web Services and Primitives of SOA  6 hours
The Web Services framework, Services (as Web Services), Service Description (with WSDL), Messaging (with SOAP)

Web Services and Contemporary SOA  12 Hours
Message Exchange patterns, Service Activity; Coordination, Atomic Transactions, Business Activities, Orchestration, Choreography, Addressing, Reliable Messaging, Correlation, Policies, Meta data Exchange, Security, Notification and eventing.

Principles of Service – Orientation  7 Hours
Services- Orientation and the enterprise, Anatomy of service-oriented Architecture, Common Principles of Service Orientation; How Service Orientation principles inter relate, Service Orientation and object orientation, Native Web Service support for service orientation principles.

Service Layers  6 Hours
Service Orientation and contemporary SOA, Service Layer Abstraction, Application service layer, Business Service Layer, Orchestration Service Layer, Agnostic Services, Service Layer Configuration scenarios.

Business Process Design  7 Hours
WS-BPEL Language basics, WS-Coordination overview, Service oriented business process redesign, WS-Addressing language basics, Ws-Reliable messaging language basics.

Enterprise Applications  8 Hours

Text Books
1. Thomas Erl: Service Oriented Architecture- Concepts, Technology and Design, Pearson Education, 2013 (listed topics only from Chapters 3,4,5,6,7,8,9,16,17)
2. Shankar Kambhapaty, Service Oriented Architecture for Enterprise and Cloud Applications, 2nd Edition, Wiley-India, 2012 (listed topics only from Chapter 5,6)

Reference Books
1. Frank cohen: FastSOA, Elsevier, 2010
Compiler Design

Subject Code: 13MCA546     IA Marks: 50
Hours/ Week: 4           Exam Hours: 3
Total Hours: 52          Exam Marks: 100

Introduction, Lexical analysis  8 Hours
Language processors; The structure of a Compilers; The evolution of programming
languages; The science of building a compiler; Applications of Compiler technology;
Programming language basics; Lexical analysis: The Role of Lexical Analyzer; Input
Buffering; Specifications of Tokens; Recognition of Tokens.

Syntax Analysis - 1  6 Hours
Introduction; Context-free Grammars; Writing a Grammar; Top-down Parsing

Syntax Analysis – 2  6 Hours
Bottom-up Parsing; Introduction to LR Parsing: Simple LR.

Syntax Analysis – 3  6 Hours
More powerful LR parsers; Using ambiguous grammars; Parser Generators.

Syntax-Directed Translation  6 Hours
Syntax-Directed definitions; Evaluation order for SDDs; Applications of Syntax-
directed translation; Syntax-directed translation schemes

Intermediate Code Generation  8 Hours
Variants of syntax trees; Three-address code; Types and declarations; Translation of
expressions; Type checking; Control flow; Back patching; Switch statements;
Intermediate code for procedures.

Run-Time Environments  6 Hours
Storage Organization; Stack allocation of space; Access to non-local data on the stack;
Heap management; Introduction to garbage collection

Code Generation  6 Hours
Issues in the design of Code Generator; The Target language; Addresses in the target code;
Basic blocks and Flow graphs; Optimization of basic blocks; A Simple Code Generator.

Text Books:
1. Alfred V Aho, Monica S. Lam, Ravi Sethi, Jeffrey D Ullman: Compilers-
   (Chapters 1, 3.1 to 3.4, 4, 5.1 to 5.4, 6, 7.1 to 7.5, 8.1 to 8.6)

Reference Books:
1. Charles N. Fischer, Richard J. IeBlanc, Jr.: Crafting a Compiler with C, Pearson
   1997.
Distributed System Models and Enabling Technologies  8 Hours

Computer Clusters for scalable parallel computing  6 Hours
Clustering for massive parallelism: Cluster Development Trends, Design Objective of Computer Clusters, Fundamental Cluster Design issues. Virtual machines and Virtualization of clusters and Data centers: Implementation levels of virtualization: levels of virtualization Implementation, VMM Design requirements and providers, Virtualization support at the OS level, Middleware Support for Virtualization.

Cloud Platform Architecture over Virtualized Data Centers  5 Hours

Public Cloud Platforms  7 Hours

Cloud Programming and Software Environments  8 Hours
Features of Cloud and Grid Platforms: Cloud Capabilities and Platform Features, Traditional Features Common to Grids and Clouds, Data Features and Databases, Programming and Runtime Support. Parallel and Distributed Programming Paradigms: Parallel Computing and Programming Paradigms, MapReduce, Twister and Iterative MapReduce, Hadoop Library from Apache.
Programming Support of App Engine 10 Hours
Programming the Google App Engine, Google File System (GFS), Bigtable, Google’s NOSQL system, Chubby, Google’s Distributed Lock service. Programming on Amazon AWS and Microsoft Azure: Programming on Amazon EC2, Amazon Simple Storage Service S3, Amazon Elastic Block Store EBS and SimpleDB, Microsoft Azure programming support. Emerging Cloud Software Environments: Open Source Eucalyptus and Nimbus, Open Nebula, Sector/Sphere, and OpenStack, Manjrasoft Aneka Cloud and Appliances.

Ubiquitous Clouds and the Internet of Things 8 Hours

Text Book:

Reference Books:
WEB 2.0 AND RICH INTERNET APPLICATIONS

SubCode: 13MCA552
Hrs/Week: 4
Total Hours: 52
IAMarks: 50
Exam Hours: 3
Exam Marks: 100

Building Rich Internet Applications with AJAX 6 Hours
Building Rich Internet Applications with AJAX: Limitations of Classic Web application model, AJAX principles, Technologies behind AJAX, Examples of usage of AJAX, Dynamic web applications through Hidden frames for both GET and POST methods. IFrames, Asynchronous communication and AJAX application model.

Ajax with XMLHTTP object 6 Hours
Creating Ajax Applications: An example, Analysis of example ajax.html, Creating the JavaScript, Creating and opening the XMLHttpRequest object, Data download, Displaying the fetched data, Connecting to the server, Adding Server-side programming, Sending data to the server using GET and POST.

Handling multiple XMLHttpRequest objects in the same page, Using two XMLHttpRequest objects, Using an array of XMLHttpRequest objects, Using inner functions, Downloading JavaScript, connecting to Google Suggest, Creating google.php, Downloading from other domains with Ajax, HTML header request and Ajax, Defeating caching, Examples.

Building XML and working with XML in JavaScript, Getting the document element, Accessing any XML element, Handling whitespace in Firefox, Handling cross-browser whitespace, Accessing XML data directly, Validating XML, Further examples of Rich Internet Applications with Ajax

Ajax Patterns 4 Hours
Predictive fetch pattern, Submission throttling pattern, Periodic refresh, Multi stage download, Fall back patterns

Working with PHP and DOM in Ajax 6 Hours
Working with PHP server variables, Getting the data in to array format, Wrapping applications in to a single PHP page, Validating input from the user, Validating integers and text, DOM, Appending new elements to a web page using the DOM and Ajax, Replacing elements using the DOM, Handling timeouts in Ajax, Downloading images with Ajax, Example programs.

Flex – 1: Understanding Flex Environment and Layouts 6 Hours
Introduction: Understanding Flex Application Technologies, Using Flex Elements, Working with Data Services (Loading Data at Runtime), The Differences between Traditional and Flex Web Applications, Understanding How Flex Applications Work, Understanding Flex and Flash Authoring. Building Applications with the Flex Framework: Using Flex
Tool Sets, Creating Projects, Building Applications, Deploying Applications.

**Flex – 2: Working with MXML and ActionScript**  
8 Hours

**Flex – 3: Working with States**  
6 Hours

**Flex – 4: Working with Data Models and Data Binding**  
4 Hours
Working with Data: Using Data Models, Data Binding, Enabling Data Binding for Custom Classes, Data Binding Examples, Building data binding proxies. Validating and Formatting Data: Validating user input, Formatting Data.

**Impacts of the Next Generation of the web**  
3 Hours
Business models for Internet and web, Data Ownership, SAAS, Socialization and cocreation of content.

**The Semantic web and Web 2.0**  
3 Hours
Overview of semantic web, Languages of the Semantic Web, Ontologies, Micro-formats, collaborative tagging and folksonomies.

**Text Books:**
1. Nicholas C Zakas et al: Professional AJAX, Wiley India, publications, (Chapters 1 to 3)

**Reference Books:**
1. Chafic Kazon and Joey Lott: Programming Flex 3, O’Reilly, 2011. (Listed topics from Chapters 1 to 8, 12 to 15)
2. Gottfried Vossen and Stephan Hagemann: Unleashing Web 2.0 Elsevier, Inc 2011 (Listed topics from Chapters 5 and 6)
UNIT 1  INTRODUCTION  
Information Retrieval, Search Engines, Search Engineers.

UNIT 2  ARCHITECTURE OF A SEARCH ENGINE  
Architecture, Basic Building Blocks, Text Acquisition, Text Transformation Index Creation, User Interaction, Ranking and Evaluation

UNIT 3  CRAWLS AND FEEDS  
Deciding what to search, Crawling the Web, Directory Crawling, Document Feeds, Conversion Problem, Storing the Documents, Detecting Duplicates, removes noise.

UNIT 4  PROCESSING TEXT  

UNIT 5  RANKING WITH INDEXES  
Abstract Model of Ranking, Inverted indexes, Compression, Entropy and Ambiguity, Delta Encoding, Bit-aligned codes, Auxiliary Structures, Index Construction, Query Processing.

UNIT 6  QUERIES AND INTERFACES  
Information Needs and Queries, Query Transformation and Refinement, Showing the Results Cross-Language Search.

UNIT 7  RETRIEVAL MODELS  
Overview of Retrieval Models, Boolean Retrieval, The Vector Space Model, Probabilistic Models, Information Retrieval as Classification, BM25 Ranking Algorithm, Complex Queries and Combining Evidence, Web Search, Machine Learning and Information Retrieval.

UNIT 8  EVALUATING SEARCH ENGINES  
The Evaluation Corpus, Logging, Effectiveness Metrics, Recall and Precision Averaging and Interpolation, Efficiency Metrics, Training, Testing, and Statistics

Text Books and References

Soft Computing

Subject Code: 13MCA554            I.A. Marks : 50
Hours/Week: 04                  Exam Hours: 03
Total Hours: 52                Exam Marks: 100

**Genetic Algorithm: An Overview** 5 Hours

**Genetic Algorithm in Problem Solving** 5 Hours
Evolving Computer Programs, Data Analysis and Prediction, Evolving Neural Networks.

**Theoretical Foundations of Genetic Algorithm** 5 Hours
Schemas and the Two-Armed Bandit Problem, Royal Roads, Exact Mathematical Models of Genetic Algorithm.

**Implementing a Genetic Algorithm** 5 Hours
When should a Genetic Algorithm be used, Encoding a Problem for a Genetic Algorithm, Adapting the Encoding, Selection Methods, Genetic Operators, Parameters for Genetic Algorithm.

**Introduction to fuzzy set theory** 8 hours
Probabilistic reasoning, Fuzzy sets, Mathematics of fuzzy set theory, Operations on fuzzy sets, Comparison of fuzzy and crisp set theory.

**Fuzzy mapping** 6 Hours
One to one mapping, Max-min principle, Extension principle, Implication rules – mamdani implications.

**Membership functions** 8 Hours
Universe of discourse, Mapping inside fuzzy domain, Fuzzy membership mapping methods, Application to real world problems.

**Neural Networks and Fuzzy System** 6 Hours
Neural and Fuzzy Machine Intelligence, Fuzziness as Multivalence, The Dynamical System Approach to Machine Intelligence: The Brain as a Dynamical System, Intelligent Behaviour as Adaptive Model Free estimation.

**Neural Network Theory** 4 Hours
Neuron as Functions, Signal Monotonicity, Biological Activations and Signals, Neuron Fields, Neuronal Dynamical System, Common Signal Functions, Pulse –Coded Signal Functions

**Text Books:**

Reference Book:
Introduction to Information Storage and Management, Storage System Environment
7 Hours

Information Storage, Evolution of Storage Technology and Architecture, Data Center Infrastructure, Key Challenges in Managing Information, Information Lifecycle Components of Storage System Environment, Disk Drive Components, Disk Drive Performance, Fundamental Laws Governing Disk Performance, Logical Components of the Host, Application Requirements and Disk Performance.

Data Protection, Intelligent Storage system
6 Hours
Implementation of RAID, RAID Array Components, RAID Levels, RAID Comparison, RAID Impact on Disk Performance, Hot Spares Components of an Intelligent Storage System, Intelligent Storage Array

Direct-Attached Storage, SCSI, and Storage Area Networks
7 Hours
Types of DAS, DAS Benefits and Limitations, Disk Drive Interfaces, Introduction to Parallel SCSI, Overview of Fibre Channel, The SAN and Its Evolution, Components of SAN, FC Connectivity, Fibre Channel Ports, Fibre Channel Architecture, Zoning, Fibre Channel Login Types, FC Topologies.

NAS, IP SAN
6 Hours

Content-Addressed Storage, Storage Virtualization
6 Hours
Fixed Content and Archives, Types of Archive, Features and Benefits of CAS, CAS Architecture, Object Storage and Retrieval in CAS, CAS Examples. Forms of Virtualization, SNIA Storage Virtualization Taxonomy, Storage Virtualizations Configurations, Storage Virtualization Challenges, Types of Storage Virtualization,

Business Continuity, Backup and Recovery
6 Hours

Local Replication, Remote Replication
7 Hours
Source and Target, Uses of Local Replicas, Data Consistency, Local Replication Technologies, Restore and Restart Considerations, Creating Multiple Replicas, Management Interface, Modes of Remote Replication, Remote Replication Technologies, Network Infrastructure.

Securing the Storage Infrastructure, Managing the Storage Infrastructure
7 Hours

**Text Books:**

**Reference Books:**
Software Project Management

Subject Code: 13MCA556           I.A. Marks : 50
Hours/Week: 4                    Exam Hours : 3
Total Hours: 52                  Exam Marks: 100

UNIT I INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT     6 Hours
Project Definition – Contract Management – Activities Covered By Software Project Management – Overview of Project Planning – Stepwise Project Planning.

UNIT II PROJECT EVALUATION          8 Hours

UNIT III ACTIVITY PLANNING         16 Hours

UNIT IV MONITORING AND CONTROL     12 Hours

UNIT V MANAGING PEOPLE AND ORGANIZING TEAMS 10 Hours

TEXT BOOK:

REFERENCES:
Software Design Laboratory

Subject Code: 13MCA56
Hours/Week : 3
Total Hours : 42

The student has to draw the necessary UML diagrams using any suitable UML Drawing Tool and implement in Java OR C++ OR C# a program to demonstrate the Design Pattern specified by the Examiner. For Analysis and Design models, diagrams such as Use-case, Class Diagram, Sequence/Collaboration Diagram Should be drawn with suitable scenario, activity diagram, component diagram & deployment diagram.

The Design Pattern is allotted based on lots from the following list:
1) Expert
2) Controller
3) Publisher-Subscriber
4) Command
5) Forward-Receive
6) Client-Dispatcher
7) Proxy
8) Façade
9) Polymorphism
10) Whole-Part
11) Master-Slave

Note: Any Supporting Tool may be used.
.NET Laboratory

Subject Code : 13MCA57    I.A. Marks : 50
Hours/Week : 3          Exam Hours: 3
Total Hours : 42        Exam Marks: 50

PART A
1. Write a Program in C# to demonstrate Command line arguments processing.
2. Write a Program in C# to demonstrate boxing and Unboxing.
3. Write a program to demonstrate Operator overloading.
4. Find the sum of all the elements present in a jagged array of 3 inner arrays.
5. Using Try, Catch and Finally blocks write a program in C# to demonstrate error handling.
6. Demonstrate Use of Virtual and override key words in C# with a simple program.
7. Write a program to demonstrate delegates.
8. Write a program to demonstrate abstract class and abstract methods in C#.
9. Write a program to illustrate the use of different properties in C#.
10. Demonstrate arrays of interface types (for runtime polymorphism) with a C# program.

PART – B
1. Consider the Database STUDENT consisting of following tables: tbl_Course (CourseID: int, CourseName: string) tbl_Student (USN: string, StudName: string, Address: string, CourseID: int, YrOfAdmsn: int)
Develop suitable windows application using C#.NET having following options:
1. Entering new course details.
2. Entering new student details.
3. Display the details of students (in a Grid) who belong to a particular course.
4. Display the details the students who have taken admission in a particular year.

2. Consider the Database BLOODBANK consisting of following tables: tbl_BloodGroup (BloodID: int, BloodGroup: string) tbl_Donor (DonorID: int, DonorName: string, Address: string, ContactNo: int, DOB: date, Gender: string, Weight: int, BloodID: int)
Develop suitable windows application using C#.NET having following options:
1. Entering Blood group details.
2. Entering new donor details.
3. Display the details of donors (in a Grid) having particular blood group.
4. Display the details of donors (in a Grid) based on gender.
5. Display the details of donors (in a Grid) based on age (above 18), weight (above 45KG) and Gender(user’s choice).

3. Consider the Database STUDENT consisting of following tables: tbl_Course (CourseID: int, CourseName: string) tbl_Book (BookID :int, BookTitle: string, Author: string, CourseID: int) tbl_Student (USN: string, StudName: string, CourseID: int) tbl_BookIssue(USN: string, BookID: int, IssueDate: Date)
Develop suitable windows application using C#.NET having following options:
1. New Course Entry.
2. New Book Entry
3. New Student Entry
4. Issue of books to a student.
5. Generate report (display in a grid) showing all the books belonging to particular course.
6. Generate report (display in a grid) showing all the books issued on a particular date.
7. Generate report (display in a grid) showing all the books issued to a particular student.

4. Develop a Web Application using C#.NET and ASP.NET for an educational institution. The master page should consist of Institution Name, Logo and Address. Also, it should provide hyperlinks to Departments, Facilities Available and Feedback. Each department page and facilities page should be designed as static pages. The hyperlinks should navigate to these static pages in the form of Content Pages associated with Master Page designed. The Feedback page should have fields to enter Name, Email and Message with Submit and Cancel Buttons. Database should be created to store these three data.

5. Develop a Web Application using C#.NET and ASP.NET for a Bank. The BANK Database should consist of following tables: tbl_Bank (BankID: int, BankName: string)
tbl_Branch (BranchID: int, BankID: int, BranchName: string) tbl_Account (AccountNo: int, BankID: int, BranchID: int, CustomerName: string, Address: string, ContactNo: int, Balance: real) (Note: AccountNo and BankID together is a composite primary key).

The master page of this web application should contain hyperlinks to New Bank Entry, New Branch Entry (of selected Bank), New Customer Entry (based on branch and bank) and Report Generation. The hyperlinks should navigate to respective content pages. These content pages provide the fields for respective data entry. The reports should be generated (display in grid) as below:
1. Display all records of particular bank.
2. Display all records of a branch of particular bank.
3. The balance should be displayed for the entered account number (Bank and Branch are input through ComboBox controls and Account number is input through TextBox).

Note:
1. **Students are required to execute one question from Part A and one from Part B.**
2. **Part A has to be evaluated for 20 marks and Part B has to be evaluated for 30 marks**
Mini Project-II Guidelines

Subject Code: 13MCA58

1. A team of only two students must develop the mini project. However during the examination, each student must demonstrate the project individually.

2. The team may implement project of their choice using any one of the technologies learnt in VSemester like OOMD, C# With.NET, using SSM concepts or any other technology along with designing of database (mandatory).

3. The team must submit a brief project report (25-30 pages) that must include the following:
   a. Introduction
   b. Literature survey
   c. Hardware & Software Requirements
   d. System Design Architecture
   e. Implementation (screenshots to be included)
   f. Testing
   g. Conclusion
   h. Future enhancements.
   j. Bibliography

   Appendix: User Manual with the help of screen shots and text description

4. The report must be evaluated for 10 Marks. Demonstration for 30 Marks and Viva for 10 Marks.
VI semester Project Work (13MCA61)

General Rules

1. Students are required to take up individual project in companies/Respective Colleges other than the mini project standards already taken up during previous semesters.

2. Project should be real time work, for total of 5 months duration

3. Project work may be application oriented or research oriented as per student and guide’s interest. Therefore the project reports will vary depending on whether it is application oriented project or research based project.

4. Regular project work weekly dairy should be maintained by the students, signed by the external guide and internal guide in order to verify the regularity of the student. (Enclosing the Format)

5. Seminars / presentation should be given at Synopsis, SRS, Design and Project Completion levels.

6. Project verification at the place of project work must mandatory by the internal guide, for completion of the work.

7. If project report is not as per the format and not a real time project, external guides will have every right to reject the project

8. Examination Marks
   - IA – 50 Marks
   - Dissertation – 125
   - Viva – 75
   - Total – 250 Marks

9. Students are encouraged and appreciated to show their project code demo along with their power point slide show during their viva-voce exams as an added advantage.

Guidelines for the Preparation of Project Reports

1. **Printing Area:** The margins should be: **Left:** 1.25”, **Right:** 1.00”, **Top and Bottom:** 1.00”. The text should be justified to occupy the full line width, so that the right margin is not ragged, with words hyphenated as appropriate. Please fill pages so that the length of the text runs to the right margin.

2. The report must be printed on one side only. Please use a high-resolution printer, preferably a laser printer with at least 300 dpi. Project reports must be printed neatly on
one side of the paper on a A4 size bond paper. The reports submitted to the
department/guide(s) must be hard bounded with dry tone Xerox.

3. **Abstract**: The abstract should summarize the contents of the report and should contain at
least 150 and at most 350 words. It should be set in 12-point font size. There should be
two blank (10-point) lines before and after the title **ABSTRACT**.

4. **Layout, Typeface, Font Sizes, and Numbering**: For the main text, please use 12-point
type and 1.5 line spacing. We recommend using **Times New Roman** fonts. Italic type
may be used to emphasize words in running text. Bold type and underlining should be
avoided.

5. **Headings**.
The chapter headings should be in capitals and must be separated from the other text by
24 point line space.

Headings should be in the form where each word is capitalized (i.e., nouns, verbs, and all
other words except articles, prepositions, and conjunctions should be set with an initial
capital) and should, with the exception of the title, be aligned to the left. The font sizes
are given in Table 1.

Here are some examples of headings: “Criteria to Disprove Context-Freeness of Collage
Languages”, “On Correcting the Intrusion of Tracing Non-deterministic Programs by
Software”, “A User-Friendly and Extendable Data Distribution System”, “Multi-flip
Networks: Parallelizing GenSAT”, “Self-determinations of Man”.

<table>
<thead>
<tr>
<th>Heading</th>
<th>Example</th>
<th>Font Size and Style</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Chapter 1 Introduction</td>
<td>16 Point Bold</td>
</tr>
<tr>
<td>First Level</td>
<td>1.1. Preamble</td>
<td>14 Point Bold</td>
</tr>
<tr>
<td>Heading</td>
<td>2.3.1. Mandatory or Regulatory Signs</td>
<td>12 Point Bold</td>
</tr>
<tr>
<td>Second Level</td>
<td>Stop and Give away signs</td>
<td>12 Point Bold</td>
</tr>
<tr>
<td>Heading</td>
<td><strong>Creation of database</strong></td>
<td>12 Point Bold, Italicized</td>
</tr>
</tbody>
</table>

Table 1 Font sizes of headings. Table captions should always be
positioned Above the tables. The final sentence of a table
caption should end without a period.
Figures and Photographs
Check that in line drawings, lines are not interrupted and have constant width. Grids and details within the figures must be clearly readable and may not be written one on top of the other. The lettering in figures should have a height of 2 mm (10-point type). Figures should be scaled up or down accordingly.
Figures should be numbered and should have a caption which should always be positioned under the figures, in contrast to the caption belonging to a table, which should always appear above the table. Please center the captions between the margins and set them in 9-point type (Fig. 1 shows an example). The distance between text and figure should be about 12 point spacing, the distance between figure and caption about 6 point spacing.

6. Formulas
Displayed equations or formulas are centered and set on a separate line (with an extra line or halfline space above and below). Displayed expressions should be numbered for reference. The numbers should be consecutive within each section or within the contribution, with numbers enclosed in parentheses and set on the right margin. For example, A correlation matrix $C_{M \times M}^f$ is computed using equation (5),

$$C_{ij}^f = \frac{d_{ij}(\Sigma A B) - (\Sigma A)(\Sigma B)}{\sqrt{d_{ij}(\Sigma A^2) - (\Sigma A)^2} \sqrt{d_{ij}(\Sigma B^2) - (\Sigma B)^2}}$$

(5)

Here, $A$ and $B$ are the feature vectors of query examples $q_i$ and $q_j$ and $d^f_{\text{ij}}$ is the dimension of feature $f_{\text{ij}}$.

7. Program Code
Program listings or program commands or algorithms in the text are normally set in typewriter font, e.g., CMTT10 or Courier.
Example of an Algorithm is

Algorithm-1: Database Creation (Mean and Standard Deviation based approach)
**Input:** Static images of potential traffic sign  
**Output:** Database created.

**Methodology:**  
For each input image do:  
- **Step1:** Preprocess the image as explained in section 4.3.1  
- **Step2:** Calculate the number of components in a sign as explained in section 4.3.1.  
- **Step3:** Calculate a feature vector as mentioned in section 4.3.2.1.  
- **Step4:** Store the feature vector computed in step 3 in the corresponding database, based on number of components present in the sign. For End.

Algorithm End.

---

8. **Footnotes/Header**  
Footnotes/Header should appear at the bottom of the normal text area, with a line of about 5 cm in Word set immediately below/above the text.

Header sample: (Project title is left aligned and page number is right aligned)  

```
<<Project Title>>     <<Page Number>>
```

**Sample Footer:**

```
<College Name>  Department of MCA  2013-2014
```

9. The list of references is headed “References” and is assigned a number with square brackets in the decimal system of headings. The list should be set in small print and placed at the end of the dissertation, in front of the appendix, if any exists. Please do not insert a page break before the list of references if the page is not completely filled. An example is given at the end of this information sheet. For citations in the text please use square brackets and consecutive numbers: [1], [2], [3] etc.

---

10. **Page Numbering**  
Reports must be printed with page numbers on the top right corner.

11. The total number of reports to be prepared are three  
- One copy to the concerned guide  
- One copy for University  
- One copy to candidate  
- Two CD’s having soft copy of Project report (for department purpose)

12. Before taking the final printout, the approval of the concerned guide is mandatory and suggested corrections, if any, must be incorporated.

13. Every copy of the report must contain (See formats towards the end of this document)  
- Outer title page (parrot green) with a plastic cover  
- Inner title page (White)  
- Certificate in the format enclosed, only certificate will be signed by following:  
  - Principal
14. The organization of the report should be as follows
- Inner title page
- Certificate
- Project Completion certificate from Company / College
- Declaration (by student)
- Acknowledgement
- Abstract
- Table of Contents
- List of table and figures
- Main body of project

Proper attention is to be paid to the technical contents as well as to the organization of the report and clarity of the expression. Care should be taken to avoid spelling and typing errors. The student should note that report (write-up) forms the important component in the overall evaluation of the project.

Sample content (more suitable for Application oriented projects) is attached and number of pages may be 40-70 which can be modified as per guide’s instructions depending on the project under development. The respective guides can decide how the content of the project report must be organized if the project is research oriented, as a specific format cannot be defined for various domains of research problems.
CONTENTS (For Application oriented Projects)

1. INTRODUCTION
   1.1 PROJECT DESCRIPTION (2-4 pages) 0
   1.2 COMPANY PROFILE (1-2-3 pages) 0

2. LITERATURE SURVEY
   2.1 EXISTING AND PROPOSED SYSTEM (2-3 pages) 0
   2.2 FEASIBILITY STUDY (2-3 pages) 0
   2.3 TOOLS AND TECHNOLOGIES USED (2-4 pages) 0
   2.4 HARDWARE AND SOFTWARE REQUIREMENTS (1 page) 0

3. SOFTWARE REQUIREMENTS SPECIFICATION
   3.1 USERS (2-3 pages) 0
   3.2 FUNCTIONAL REQUIREMENTS (2-3 pages) 0
   3.3 NON-FUNCTIONAL REQUIREMENTS (2-3 pages) 0

4. SYSTEM DESIGN (High level or Architectural design)
   4.1 SYSTEM PERSPECTIVE (1-2 pages) 0
   4.2 CONTEXT DIAGRAM (1-2 pages) 0

5. DETAILED DESIGN (various design diagrams according to project)
   5.1 USE CASE DIAGRAM (4-6 pages) 0
   5.2 SEQUENCE DIAGRAMS (4-6 pages) 0
   5.3 COLLABARATION DIAGRAMS (3-5 pages) 0
   5.4 ACTIVITY DIAGRAM (4-6 pages) 0
   5.5 DATABASE DESIGN (ER and/or Conceptual schema) (3-4 pages) 0

6. IMPLEMENTATION (no full code, code snippet may be included)
   6.1 SCREEN SHOTS (15-20 pages) 0

7. SOFTWARE TESTING (Test cases etc.) (6-8 pages) 0
8. CONCLUSION (1 page) 0
9. FUTURE ENHANCEMENTS (1 page) 0

Appendix A BIBLIOGRAPHY (1 page) 0
Appendix B USER MANUAL (2-10 pages) 0
PROJECT TITLE

A Dissertation submitted in partial fulfillment of the requirements for the award of degree of

MASTER OF COMPUTER APPLICATIONS

of

Visvesvaraya Technological University

By

STUDENT NAME

(USN Number)

Under the Guidance of

GUIDE NAME

Department of Master of Computer Applications, Institute Name, Address, Place, Pincode.

June 2013

(Certificate)

INSTITUTE NAME

LOGO
Department of Master of Computer Applications, Institute Name
Address

CERTIFICATE

This is to Certify that Student Name has completed his/her final semester project work entitled “******” as a partial fulfillment for the award of Master of Computer Applications degree, during the academic year 2013 under my (our joint) supervision.

Signature of Internal Guide          Signature of External Guide

Guide Name          Guide Name
Affiliation          Affiliation
Address            Address

Head of the Dept./Director          Principal

Declaration

I, <Name of student>, student of 6th MCA, <College Name>, bearing USN <USN of student> hereby declare that the project entitled <Project Title> has been carried out by me under the supervision of External Guide /(or Guide) <Name of the Guide>, <Designation of Guide> and Internal Guide (or Co-guide) <Name of the Guide>, <Designation of Guide> and submitted in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications by the Visvesvaraya Technological University during the academic year 2014. This report has not been submitted to any other Organization/University for any award of degree or certificate.

Name:

Signature:
External Examiner Project Dissertation Evaluation Form

Project Title:

Student Name With USN:

1. Relevance of the project with respect to real scenario: (External Examiner will have to write maximum of 5 lines for this) – 50 Marks

2. Project Standard with respect to the post graduation / research / application oriented, to be explained by the external examiner – 50 Marks

3. Final Testing and Results with technical conclusions – 25 Marks

Total Marks:

Signature of the External Examiner with Name and Date