Data Structures Using C

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I

Subject Code	16MCA11	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand basics of C programming language

CO2: Acquire knowledge of

- Various types of data structures, operations and algorithms
- Sorting and searching operations

CO3: Analyze the performance of

- Stack, Queue, Lists, Trees, Hashing, Searching and Sorting techniques

CO4: Implement all the applications of Data structures in a high-level language

CO5: Design and apply appropriate data structures for solving computing problems.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Overview of C 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.**Functions**: Categories of functions, Call by Value, Call by reference. **Arrays*:*** Definition, Representation, Single dimension, Two dimensional, Multi-dimensional Arrays, Passing arrays to functions, passing strings to functions	
Module -2	**10 Hours**
Pointers, Pointer Expression, Pointer as function arguments, Functions returning pointers, Pointers to functions, **Structures:** Declaring and using structure types. **Classification of Data Structures**:Primitive and Non- Primitive, Linear and Nonlinear; Data structure Operations, ADT, Array as ADT, Operations - Insert, Delete, Search, Sort, **String** Definition, Representation, String as ADT, Operations – Insert, Delete, Concatenate, Comparing, Substring.	
Module -3	**10 Hours**
Stack*:* Definition, Representation, Stack as ADT, Operations and Applications: Polish and reverse polish expressions, Infix to postfix conversion, evaluation of postfix expression, infix to prefix, postfix to infix conversion; Recursion - Factorial, GCD, Fibonacci Sequence, Tower of Hanoi **Queue*:*** Definition, Representation, Queue as ADT, Operations, Queue Variants: Circular Queue, Priority Queue, Double Ended Queue; Applications of Queues. Programming Examples.	
Module -4	**10 Hours**
Linked List: Limitations of array implementation, Memory Management: Static (Stack) and Dynamic (Heap) Memory Allocation, Memory management functions. Definition, Representation, Operations: getnode() and Freenode() operations, Types: **Singly Linked List**. Linked list as a data Structure, Inserting and removing nodes from a list, Linked implementations of stacks, Example of list operations such as insert and	

delete an element before a key element, Header nodes, Array implementation of lists.: **Circular Linked List**: Inserting, deleting and searching elements in a lists, **Double Linked List**: Inserting and Deleting Nodes, Queue as doubly linked lists, such as insert into position, Delete an specified element. **Application of Linked Lists**: Stacks, Queues, Double-ended Queues, Priority Queues, Sparse Matrix and Polynomials using Lists, Trees, BST.	
Module -5	**10 Hours**
Trees: Definitions, Terminologies, Array and linked Representation of Binary Trees, Types- Complete/full, Almost Complete, Strictly, Skewed; Traversal methods - Inorder, postorder, preorder; Binary Search Trees - Creation, Insertion, Deletion, Traversal, Searching. **Sorting & Searching**: Bubble sort, Insertion Sort, Selection sort, Quick sort, Linear Search, Binary Search and BST. **Hashing**: The Hash Table organizations, Hashing Functions, Static and Dynamic Hashing, Collision-Resolution Techniques, Programming Examples.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 questions from each module.
- Each question will have questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. **Programming in ANSI C, Balaguruswamy,** McGraw Hill Education
2. **Data Structures Using C and C++** by YedidyahLangsam and Moshe J. Augenstein and Aaron M Tenanbanum, 2nd Edition, Pearson Education Asia, 2002.
3. **Introduction to Data Structure and Algorithms with C++** by Glenn W. Rowe

Reference Books:

1. **Principles of Data Structures using C & C++** by Vinu V. Das, New Age International, 2006
2. **Data Structures Using C , Balaguruswamy:,** McGraw Hill Education

UNIX Programming
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – I

Subject Code	16MCA12	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand and experience the UNIX environment, File system and hierarchy.
CO2: Demonstrate commands to extract, interpret data for further processing.
CO3: Apply commands to perform different tasks on various applications
CO4: Analyze the usage of different shell commands, variables and AWK filtering.
CO5: Evaluate different commands with sample shell scripts

Modules	**Teaching Hours**
Module -1	**10 Hours**
Introductionof UNIX and Shell: Introduction, History, Architecture, Experience the Unix environment, Basic commands ls, cat, cal, date, calendar, who, printf, tty, sty, uname, passwd, echo, tput, bc, script, spell and ispell, 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.	
Module -2	**10 Hours**
UNIX File System: The file, what's in a filename? The parent-child relationship, pwd, the Home directory, absolute pathnames, using absolute pathnames for a command, cd, mkdir, rmdir, Relative pathnames, The UNIX file system. 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.	
Module -3	**10 Hours**
Simple Filters: Pr, head, tail, cut, paste, sort, uniq, tr commands, Filters using Regular Expression : grep&sedgrep, 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.	
Module -4	**10 Hours**
Awk and Advanced Shell Programming: 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. The sh command, export, cd, the Command, expr, Conditional Parameter Substitution, Merging Streams, Shell Functions, eval, Exec Statement and Examples	
Module -5	**10 Hours**
Process and System Administration: 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.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Your UNIX-The Ultimate Guide, Sumitabha Das, Tata McGrawHill,

Reference Books:

1. "Unix Shell Programming", YashwantKanetkar,
2. "Beginning Shell Scripting", Eric Foster-Johnson, JohnCWelch, MicahAnderson, Wroxpublication.
3. UNIX: Concepts and Applications, Sumitabha Das, Tata McGrawHill,
4. "Introduction to UNIX" by M.G.Venkatesh Murthy.

Web Technologies

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – I

Subject Code	16MCA13	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand the fundamentals of web and thereby develop web applications using various development languages and tools.
CO2:Build the ability to select the essential technology needed to develop and implement web applications
CO3: Use Scripting language utilities for static and dynamic environment
CO4: Design XML document with presentation using CSS and XSLT.
CO5: Develop CGI applications using PERL.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Web Fundamentals Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, the Web Programmers Toolbox. Evolution of the Web, Peak into the History of the Web, Internet Applications, Networks, TCP/IP, Higher Level Protocols, Important Components of the Web, Web Search Engines, Application Servers	
Module -2	**10 Hours**
Introduction to XHTML and CSS Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links. Lists, Tables, Forms, Frames, syntactic differences between HTML and XHTML. **Cascading Style Sheets:** Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The box model, Background images, The <span> and <div> tags, Conflict resolution.	
Module -3	**10 Hours**
The basics of JavaScript Overview of JavaScript, Object orientation and JavaScript, general 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**,** **JavaScript and XHTML Documents** The JavaScript Execution Environment, The Document Object Model, Elements Access in Java Script, Events and Event Handling, Handling Events from Body Elements, Handling Events from Text Box and password Elements, The DOM2 Model, The navigator Object, Dom Tree Traversal and Modification.	
Module -4	**10 Hours**
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 Introduction, Syntax of XML, XML Document Structure, Document type definitions, Namespaces, XML schemas, displaying raw XML documents, Displaying XML documents with CSS, XSLT style sheets, XML processors, Web services.	
Module -5	**10 Hours**
Perl and CGI Programming 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.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Robert W.Sebesta : Programming the World Wide Web, 4thEdition, Pearson education, 2012. **(Chapters 1, 2, 3, 4, 5, 6, 7, 8, 9)**
2. M.Srinivasan: Web Programming Building Internet Applications, 3rdEdition, WileyIndia, 2009. **(Chapter 1)**

Reference Books:

1. Jeffrey C.Jackson: Web Technologies-A Computer Science Perspective, Pearson Education, 7th Impression, 2012.
2. Chris Bates: Web Technology Theory and Practice, Pearson Education, 2012.
3. Internet Technology and Web Design, Instructional Software Research and Development(ISRD) Group, TataMcGrawHill,2011 Raj Kamal : Internet and Web Technologies, McGraw Hill Education

Computer Organization

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I

Subject Code	16MCA14	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand the Basics of Digital System

CO2: Understand the Basics of Computer System Organization

CO3: Apply the concepts of the number system in Designing Digital System.

CO4: Analyze the need of Logic circuits in digital system

CO5: Create logic circuits for real time requirement

Modules	**Teaching Hours**
Module -1	**10 Hours**
Binary Systems and Combinational Logic DigitalComputersandDigitalSystems,BinaryNumbers,NumberBaseConversion,OctalandHexadecimalNumbers,subtractionusingr’sandr-1complements,BinaryCode,BinaryStorage andRegisters,BinaryLogic, IntegratedCircuits. AxiomaticDefinitionofBooleanAlgebra,BasicTheoremsandPropertiesofBooleanAlgebra,Boolean Functions,CanonicalandStandardForms,OtherLogicOperations,DigitalLogicGates,ThemapMethod,Two–andThree–VariableMaps,Four–VariablesMap	
Module -2	**10 Hours**
Arithmetic Circuits and Sequential Logic NANDandNORImplementation,OtherTwo-LevelImplementations,Don’tCareConditions.Introduction,Adders, Subtractors,BinaryParallelAdder,DecimalAdder,MagnitudeComparator,Decoders, Multiplexers,BOOTHalgorithmforsignednumberswithexample. SequentialLogic: Introduction,differenttypesofFlip–Flops,TriggeringofFlip-Flops,Registers,ShiftRegisters,RipplecounterandSynchronousCounter.	
Module -3	**10 Hours**
Computer Types, Functional Units, Basic Operational Concepts, Bus structure, Software, Performance, Multiprocessing and Multi computers, Introduction to Assemblers and Compilers. Machine Instruction: Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes.	
Module -4	**10 Hour**

	s
Assembly Language and Input /Output Organization Basics of Assembly Language Programme, ExamplesfromAssemblyLanguageProgramming. AccessingI/ODevices,Interrupts,DMA,ProcessorExample,Buses.CasestudyofIA32Intelprocessor.	
Module -5	**10 Hours**
The Memory System BasicConcepts,SemiconductorRAMMemories,Read–OnlyMemories,Speed,Size,andCost,CacheMemories,VirtualMemories,MemoryManagementRequirements,SecondaryStorage.	
Question paper pattern: • The question paper will have ten questions. • Each full question consists of 16 marks. • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. M.Morris Mano, “Digital Logic and Computer Design”, Pearson, 2012. 2. CarlHamacher, ZvonkoVranesicSafwatZaky, ”Computer Organization”, 5thedition, TataMcGraw-Hill, 2011	
Reference Books: 1. JohnP.Hayes, “Computer Architecture and Organization”, Tata McGraw-Hill, Edition, 2012. 2. SoumitrsKumar Mandal,“Digital ElectronicsPrinciples and Applications”, Tata McGraw-Hill, 2010 3. Hamacher , “ Computer Organization” , McGraw-Hill Education	

DISCRETE MATHEMATICAL STRUCTURES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – I

Subject Code	16MCA15	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Use the logical notation to define and reason about fundamental mathematical concepts such as sets, relations, functions, and integers.

CO2: Calculate numbers of possible outcomes of elementary combinatorial processes such as permutations and combinations.

CO3: Calculate probabilities and conditional probabilities.

CO4: Apply graph theory models of data structures and state machines to solve problems of connectivity and constraint satisfaction, for example, scheduling.

Modules	Teaching Hours
Module -1	**10 Hours**
Logic: Propositional logic, equivalences, predicates and quantifiers, rules of inference, introduction to proofs, proof methods.	
Module -2	**10 Hours**
Sets, Functions and Relations: Sets, set operations, Functions, Relations, equivalence relations and partial ordering.	
Module -3	**10 Hours**
Counting: Basics of counting, the pigeonhole principle, permutations and combinations, Binomial Co-efficients, recurrence relations.	
Module -4	**10 Hours**
Probability: Introduction to probability, axioms of probability, independence and conditional probability, inclusion-exclusion principle.	
Module -5	**10 Hours**
Graph Theory: Graphs, terminology and special types of graphs, representation of graphs,	

isomorphism, connectivity, Euler and Hamiltonian paths, shortest path problems, planar graphs, graph coloring.

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Discrete Mathematics and its Applications by Kenneth H Rosen, 7th edition, (Indian adaptation by Kamala Krithivasan), Tata McGraw Hill, 2011.
2. Discrete and Combinatorial Mathematics: An Applied Introduction by Ralph P. Grimaldi and B V Ramana (Indian corrupted adaptation), 5th edition, Pearson, 2011.

Reference Books:

1. Graph Theory with Applications to Engineering and Computer Science by NarsinghDeo, Prentice Hall India, 2004.
2. Discrete Mathematical Structures with Applications to Computer Science by J. P. Tremblay and R. Manohar, McGraw Hill.

Note: Unit IV to be covered from Text Book 2.
Module I: T1: 1.1 – 1.3, 1.5,1.6
Module II: T1: 2.1- 2.3, 7.1, 7.5, 7.6
Module III: T1:5.1 – 5.4, 6.1, 6.2
Module IV: T2: 3.4 – 3.6, T1: 6.5
Module V: T1: 8.1 – 8.8

DATA STRUCTURES USING C LABORATORY

Subject Code	16MCA16	CIE Marks	20
Number of Lecture Hours/Week	01Hour Tutorial/ Instructions 02 Hours Laboratory	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03

CREDITS – 02

Course Outcomes(CO):At the end of this course, the students will be able to

*CO1:*Apply data structure concepts to develop interactive applications in C.

CO2: Linear data structures and their applications such as Stacks, Queues and Lists

CO3: Non-Linear Data Structures and their Applications

CO4: Be fluent in the use of different types of sorting and searching techniques

Laboratory Experiments:

1. Write a menu driven Program in C for the following **Array operations**
   a. Creating an Array of **N** Integer Elements
   b. Display the Array Elements
   c. Inserting an Element at a given valid Position
   d. Deleting an Element at a given valid Position
   e. Exit.
2. Write a C program to Implement the following **searching techniques**
   a. Linear Search
   b. Binary Search
3. Write a C program to implement the following **sorting** algorithms using user defined functions:
   a. Bubble sort (Ascending order)
   b. Selection sort (Descending order)
4. Write a Program in C for the following **String operations** (without using built-in functions)
   a. Read a main String (**STR),** a Pattern String (**PAT**) and a Replace String (**REP**)
   b. Perform Pattern Matching Operation: Find and Replace all occurrences of **PAT** in **STR**

with **REP** if **PAT** exists in **STR.** Report suitable messages in case **PAT** does not exist in**STR** Support the program with functions for each of the above operations.

5. Write a C Program to create a class called STACK to store Integers for the following operations (**Array Implementation of Stack** with maximum size **MAX**)
   a. ***Push*** an Element on to Stack
   b. ***Pop*** an Element from Stack

c. Demonstrate ***Overflow*** and ***Underflow*** situations on Stack

d. Display the status of Stack

e. Exit

6. Implement a Program in C for converting an **Infix Expression to Postfix Expression**.

7. Implement a Program in C for evaluating an **Postfix Expression**.

8. Write a C program to implement the following using **recursion**

   a. Sum of n numbers

   b. Generate Fibonacci sequence

   c. Solve Towers of Hanoi Problem

9. Implement a menu driven Program in C for the following operations on **Circular QUEUE** of Integers (Array Implementation of Queue with maximum size **MAX**)

   a. Insert an Element on to Circular QUEUE

   b. Delete an Element from Circular QUEUE

   c. Demonstrate ***Overflow*** and ***Underflow*** situations on Circular QUEUE

   d. Display the status of Circular QUEUE

   e. Exit

10. Write a program to Simulate the working of a dequeue

11. Implement a menu driven Program in C for the following operations on **Singly Linked List (SLL)** of Student Data with the fields: ***USN, Name, Branch, Sem***

   a. Create a **SLL** of **N** Students Data.

   b. Display the status of **SLL** and count the number of nodes.

   c. Perform Insertion at the beginning /end of **SLL.**

d. Perform Deletion at the beginning /end of **SLL.**

e. Exit.

12. Write a program to Simulate the working of a **Singly circular linked list** providing the following operations

   a. Delete from the beginning/end

   b. Delete a given element

   c. Display &Insert is mandatory

13. Write a C Program using **Doubly Linked List** to Implement Stack operations to store Integers

   a. ***Push*** an Element on to Stack

   b. ***Pop*** an Element from Stack

d. Demonstrate ***Overflow*** and ***Underflow*** situations on Stack

   e. Display the status of Stack

   f. Exit

14. Implement a menu driven Program in C for the following operations on **Binary Tree** of Integers

   a. Create a BST of **N** Integers

   b. Traverse the BST in Inorder, Preorder and Post Order

   c. Search the BST for a given element and report the appropriate message

   d. Delete an element from BST

   e. Exit

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfroma lot of all14 questions.

Note 2: Change of program is not permitted in the Practical Examination.

UNIXProgrammingLaboratory			
Laboratory Code	16MCA17	CIE Marks	20
Number of Lecture Hours/Week	01Hour Tutorial/ Instructions 02 Hours Laboratory	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03
CREDITS – 02			
Course Outcome (CO): At the end of this course, the students will be able to **CO1:** Understand the Unix programming environment. **CO2:** Be fluent in the use of Vi editor. **CO3:** Be able to design and implement shell scripts to manage users with different types of permission and file based applications. **CO4:** Be fluent to write Awk scripts.			
Laboratory Experiments:			
A. ExploretheUnixenvironment.			
B. Explorevieditorwithvimtutor.Performthefollowingoperationsusingvieditor,butnotlimitedto: 1. Insert character, delete character,replace character 2. savethefileandcontinueworking 3. save thefileand exittheeditor 4. quittheeditor 5. quitwithoutsavingthefile 6. rename afile 7. insertlines,deletelines, 8. setlinenumbers 9. searchforapattern 10. moveforwardandbackward			
1a. Writeashellscriptthattakesavaliddirectorynameasanargumentandrecursivelydescendallthesub-directories,findsthemaximumlengthofanyfileinthathierarchyandwritesthismaximumvaluetothestandard output. **1b.**Writeashellscriptthatacceptsapathnameandcreatesallthecomponentsinthatpathnameasdirectories.Forexample, if the scriptis namedmpc,thenthe commandmpca/b/c/dshouldcreate directoriesa,a/b,a/b/c,a/b/c/d.			
2a.Writeashellscriptthatacceptstwofilenamesasarguments,checksifthepermissionsforthesefilesareidenticalandifthepermissionsareidentical,outputcommonpermissionsandotherwiseoutputeach			

filenamefollowedbyitspermissions.

2b.Writeashellscriptwhichacceptsvalidlog-innamesasargumentsandprintstheircorresponding homedirectories,ifno argumentsarespecified,printasuitableerrormessage.

3a. Createascriptfilecalledfile-propertiesthatreadsafilenameenteredandoutputsitProperties

3b. Writeshellscripttoimplementterminallocking(similartothelockcommand).Itshouldprompttheuserfor apassword.Afteracceptingthepasswordenteredbytheuser,itmustpromptagainforthematchingpasswordasconfirmationandifmatchoccurs,itmustlockthekeyworduntilamatching passwordisenteredagainbytheuser,Notethatthescriptmustbewrittentodisregard BREAK,control-D.Notime limitneedbe implementedforthe lockduration.

4a. Writeashellscriptthatacceptoneormore filenamesasargument and convertallofthemto uppercase,providedtheyexistincurrentdirectory.

4b. Writeashellscriptthatdisplaysallthelinkstoafilespecifiedasthefirstargument tothescript.Thesecondargument,whichisoptional,canbeusedtospecifyinwhichthesearchis tobegin.Ifthissecondargumentisnotpresent,thesearchistobeginincurrent workingdirectory.Ineithercase,thestartingdirectoryaswellasallitssubdirectoriesat alllevelsmustbesearched. Thescript neednotincludeanyerrorchecking.

5a. Writeashellscriptthatacceptsasfilenameasargumentanddisplayitscreationtimeiffileexistandifitdoesnotsendoutputerror message.

5b. Writeashellscriptto displaythecalendar for currentmonth withcurrentdatereplacedby *or**dependingonwhetherthedatehasone digitortwodigits

6a. Writeashellscripttofindafile/sthatmatchesapatterngivenascommandlineargumentinthehomedirectory,displaythecontentsofthefileandcopythefileintothedirectory ~/mydir

6b. Writeashellscripttolistallthefilesinadirectorywhosefilenameisatleast10characters.(useexprcommandtocheckthelength)

7a. Writeashellscriptthatgetsexecuteddisplaysthemessageeither“GoodMorning”or“GoodAfternoon”or “GoodEvening”dependingupontimeatwhichtheuserlogsin.

7b. Writeashellscriptthatacceptalistoffilenamesasitsargument,countandreportoccurrenceof eachword thatispresentinthefirstargumentfileon other argumentfiles.

8a. Writeashellscriptthatdeterminetheperiodforwhichaspecifieduserisworkingonsystemanddisplayappropriatemessage.

8b. Writeashellscriptthatreportstheligginginofaspecifieduserwithinoneminuteafterhe/shelogin.Thescriptautomaticallyterminateifspecifieduserdoesnotloginduringaspecifiedperiodof time.

9a. Writeashellscriptthatacceptthefilename,startingandendinglinenumberasanargumentand displayallthelines betweenthegivenlinenumber.

9b.
Writeashellscriptthatfoldslonglinesinto40columns.Thusanylinethatexceeds40charactersmustbebroken after40th,a“\”istobeappendedastheindicationoffoldingandtheprocessingistobecontinuedwiththeresidue.Theinputistobesuppliedthroughatextfilecreatedbytheuser.

10a. Write an awk scriptthatacceptsdateargumentin theform ofdd-mm-yyand displaysitintheform ifmonth,dayandyear.Thescriptshouldcheckthevalidityof theargumentandinthecaseoferror,displayasuitable message.

10b. Writeanawkscripttodelete duplicatedline from a text file.Theorderof theoriginallinesmustremainunchanged.

11a. Writeanawkscripttofindouttotalnumberofbookssoldineachdisciplineaswellastotalbook soldusingassociate arraydowntableasgivenbelow.

Electrical	34
Mechanical	67
Electrical	80
ComputerScience	43
Mechanical	65
Civil	98
ComputerScience	64

11b. Writeanawkscripttocomputegrosssalaryofanemployeeaccordinglytorulegivenbelow.

If basicsalaryis<10000thenHRA=15%of basic&DA=45%ofbasicIfbasicsalaryis>=10000thenHRA=20%ofbasic&DA=50% ofbasic.

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfroma lot of all1-11 questions.

Note 2: Change of program is not permitted in the Practical Examination.

WEB PROGRAMMING LABORATORY

Laboratory Code	16MCA18	CIE Marks	20
Number of Lecture Hours/Week	01Hour Tutorial/ Instructions 02 Hours Laboratory	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03

CREDITS – 02

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand the concept and usages web based programming techniques.

CO2: Learning and developing XHTML documents using JavaScript and CSS

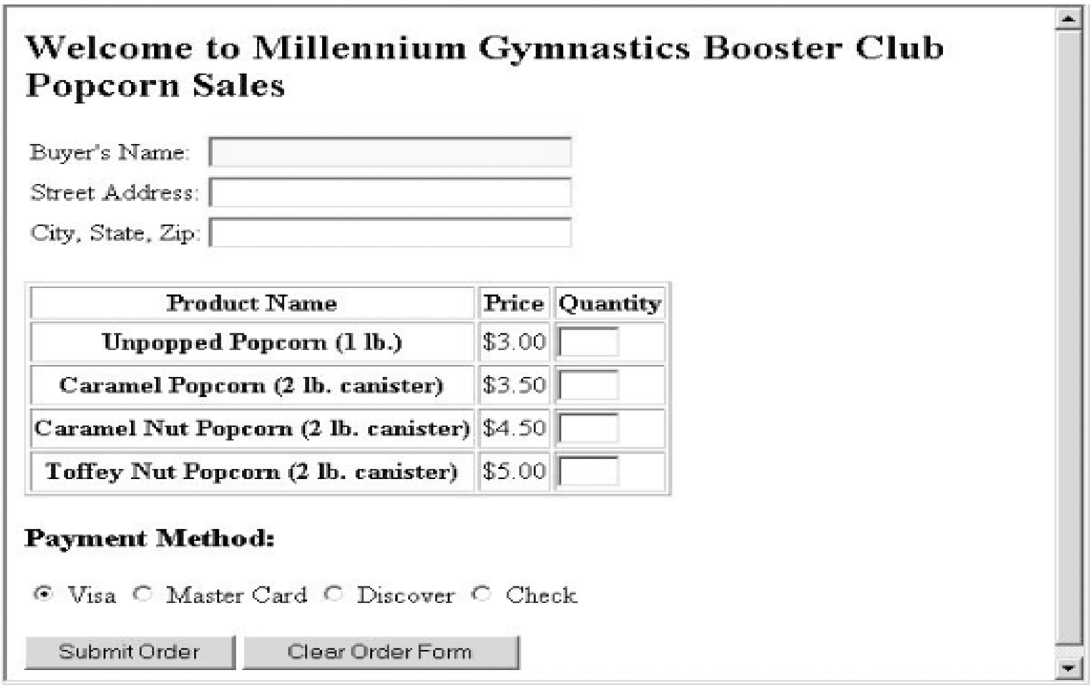
CO3: To be familiar in the use of CGI and Perl programs for different types of server side applications.

CO4: Design and implement user interactive dynamic web based applications.

Laboratory Experiments:

1. Create an XHTML page that provides information about your department. Your XHTML page must use the following tags:
   a. Text Formatting tags
   b. Horizontal rule
   c. Meta element
   d. Links
   e. Images
   f. Tables
   (If needed use additional tags).
2. Develop and demonstrate the usage of inline, external and internal style sheet using CSS. Use XHTML page that contains at least three paragraphs of text, listed elements and a table with four rows and four columns.
3. Create a XHTML document that describes the form for taking orders for popcorn. Text boxes are used at the top of the form to collect the buyer's name and address. These are placed in a borderless table to force the text box align vertically. A second table to collect actual order. Each row of this table names a product, displays the price, and uses text box with size 2 to collect the quantity ordered using <td> tag. The payment method is input by the user through one of four radio buttons. Provide provision for submission of order and clear the order form.

   Sample output form



Welcome to Millennium Gymnastics Booster Club Popcorn Sales

Buyer's Name:

Street Address:

City, State, Zip:

Product Name	Price	Quantity
Unpopped Popcorn (1 lb.)	$3.00	
Caramel Popcorn (2 lb. canister)	$3.50	
Caramel Nut Popcorn (2 lb. canister)	$4.50	
Toffey Nut Popcorn (2 lb. canister)	$5.00	

Payment Method:

● Visa ○ Master Card ○ Discover ○ Check

4. Write a JavaScript program to generate n number of random numbers and store them in an array. Sort the generated numbers in ascending order using array sort method. Develop separate functions to find mean and median of numbers that are in the array. Display the results with appropriate messages.

5. Develop, test and validate an XHTML document that has checkboxes for apple (59 cents each), orange (49 cents each), and banana (39 cents each) along with submit button. Each checkboxes should have its own ***onclick*** event handler. These handlers must add the cost of their fruit to a total cost. An event handler for the ***submit*** button must produce an alert window with the message '***your total cost is $xxx'***, where xxx is the total cost of the chose fruit, including 5 percent sales tax. This handler must return 'false' (to avoid actual submission of the form data).

Modify the document to accept quantity for each item using textboxes.

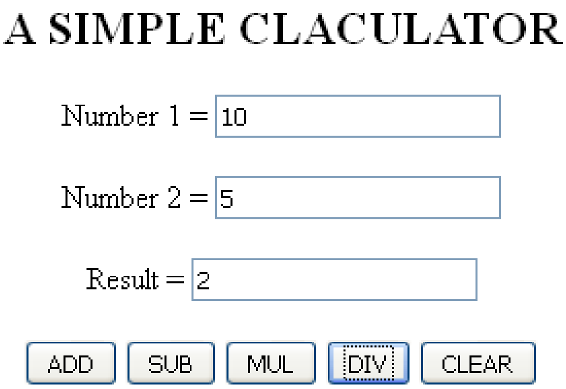
6. a) Develop and demonstrate, a XHTML document that collects the USN(the valid format is : A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by three upper-case characters followed by two digits; (no embedded spaces are allowed) from the user. Use JavaScript that validate the content of the document. Suitable messages should be display in the alert if errors are detected in the input data. Use CSS and event handlers to make your document good-looking and effective.

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

7. Develop and demonstrate, using JavaScript script, a XHTML document that contains three short paragraphs of text, 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.

Modify the above document so that when a text is movedfrom the top stacking position, it returns to its original position rather than to the bottom.

8. Develop a simple calculator to perform arithmetic (addition, subtraction, multiplication and division) operations on given two numbers. Use an html tag that allows the user to input two numbers and to display the result of arithmetic operation. Write suitable HTML and JavaScript and CSS to your simple calculator. The following figure show sample document display.



A SIMPLE CLACULATOR

Number 1 = 10

Number 2 = 5

Result = 2

Modify your program to make HTLM document as eye-catching using CSS.

9. a. Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, and Name of the College, Brach, 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.

10. Write a Perl program to process a file which contains English words, where each word is separated from the next word on a line by one space. The file is specified on the command line. The output of your program is a table in which the first column has unique words from the input file and second the second column has the number of times the word appeared in the file; no word can appear twice in the table. Use two arrays to store the table, one for the words and one for the frequency values.

Modify the program to count number of characters, words, and lines in the specified file

11. A file contains lines of employee data, where each line has name:age:deparmentcode:salary. Write a Perl program to generate the following output:

1. The names of all the employee whose names end with “son”
2. Percentage of employees under 40 years old
3. Average salary of employees under 40 years old
4. An alphabetical list of employees who are under 40 years old and who have salaries more than $40,000.

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

Modify this Perl program to count the number of visitors visiting this web page and display that number of times each visitor is visited.

13. Write an XHTML document to create a form with the following capabilities:

a. A text widget to collect the user’s name

b. B four checkboxes, one each for the following items

I. Four 100-watt light bulbs for $2.39

II. Eight 100-watt light bulbs for $4.29

III. Four 100-watt long-life light bulbs for $3.95

IV. Eight 100-watt long-life light bulbs for $7.49

c. A collection of three radio buttons they are labeled as follows:

I. Visa

II. MasterCard

III. Discover

Write a Perl CGI program that computes the total cost of the ordered light bulbs by a customer. The program must inform the buyer of exactly what was ordered in a table.

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. Modify this program to return the number of months, days, hours, and minutes since last visit by the current client. Your display should include current visit day, date and time, last visit day, date, and time, & elapsed time.

The sample out is as follows:

```
User name: XXXXXXXXXXX
Current visit time : Friday 05/08/2016, 9:30:30 AM
Last visit time    : Thursday 04/08/2016, 3:20:00 PM
Elapsed time :   18:10:30
```

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfroma lot of all14 questions.

Note 2: Change of program is not permitted in the Practical Examination.

PYTHON PROGRAMMING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – II

Subject Code	16MCA21	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand and comprehend the basics of python programming.

CO2: Apply knowledge in real time applications.

CO3: Understands about files and its applications.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Installing Python, Simple program using Python, Expressions and Values, Variables and Computer Memory, error detection, Multiple line statements, Designing and using functions, functions provided by Python, Tracing function calls in memory model, omitting return statement. Working with Text: Creating Strings of Characters, Using Special Characters in Strings, Creating a Multiline String, Printing Information, Getting Information from the Keyboard.	
Module -2	**10 Hours**
A Boolean Type , Choosing Statements to Execute, Nested If Statements , Remembering the Results of a Boolean Expression Evaluation , A Modular Approach to Program Organization, Importing Modules , Defining Your Own Modules, Testing Code Semi automatically Grouping Functions Using Methods: Modules, Classes, and Methods , Calling Methods the Object-Oriented Way, Exploring String Methods, Underscores.	
Module -3	**10 Hours**
Storing Collections of Data Using Lists: Storing and Accessing Data in Lists, modifying Lists, Operations on Lists, Slicing Lists, Aliasing, List Methods, Working with a List of Lists. Repeating Code Using Loops: Processing Items in a List, Processing Characters in Strings, Looping Over a Range of Numbers, Processing Lists Using Indices, Nesting Loops in Loops, Looping Until a Condition Is Reached, Repetition Based on User Input, Controlling Loops UsingBreak and Continue Reading and Writing.	
Module -4	**10 Hours**
Files: Kinds of files, Opening a File, Techniques for Reading Files, Files over the Internet, Writing Files, and Writing Algorithms That Use the File-Reading Techniques, Multiline Records. Storing Data Using Other Collection Types: Storing Data Using Sets, Storing Data Using Tuples, Storing Data Using Dictionaries, Inverting a Dictionary, Using the In Operator on Tuples, Sets, and Dictionaries, Comparing Collections.	
Module -5	**10 Hours**
Collection of New Information Object-Oriented Programming : Understanding a Problem Domain , Function “Isinstance,” Class Object, and Class Book , Writing a Method in Class Book, Plugging into Python Syntax: More Special Methods ,Creating Graphical User interface: Building a Basic GUI, Models, Views, and Controllers, Customizing the Visual Style Introducing few more Widgets, Object-Oriented GUIs,	

Keeping the Concepts from Being a GUI Mess.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Practical Programming: An introduction to Computer Science Using Python, second edition, Paul Gries, Jennifer Campbell, Jason Montojo, The Pragmatic Bookshelf.
2. Learning with Python: How to Think Like a Computer Scientist Paperback – Allen Downey , Jeffrey Elkner, 2015

Reference Books:

1. Introduction to Python for Computational Science and Engineering (A beginner's guide), Hans Fangohr.
2. Exploring Python, Timothy A. Budd, McGraw Hill Education
3. Python for Informatics: Exploring Information, Charles Severance.
4. Learning Python, Fourth Edition, Mark Lutz, O'Reilly publication

Object Oriented Programming Using C++

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – II

Subject Code	16MCA22	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Differentiate between object oriented programming and procedure oriented programming & Disseminate the importance of Object oriented programming

CO2: Apply C++ features such as Classes, objects, constructors, destructors, inheritance, operator overloading, and Polymorphism, Template and exception handling in program design and implementation.

CO3: Use C++ to demonstrate practical experience in developing object-oriented solutions.

CO4: Analyze a problem description and build object-oriented software using good coding practices and techniques.

CO5: Implement an achievable practical application and analyze issues related to object-oriented techniques in the C++ programming language.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Overview of C++: The Origins of C++ ,What Is Object-Oriented Programming? ,Encapsulation ,Polymorphism , Inheritance. Some C++ Fundamentals ,A Sample C++ Program ,A Closer Look at the I/O Operators, Declaring Local Variables ,No Default to int, The bool Data Type , Old-Style vs. Modern C++ , The New C++ Headers , Namespaces, Working with an Old Compiler, Introducing C++ Classes, Function Overloading, Operator Overloading ,Inheritance Constructors and Destructors ,The C++ Keywords, The General Form of a C++ Program. **Classes and Objects:** Classes, Structures and Classes Are Related, Unions and Classes Are Related , Anonymous Unions, Friend Functions, Friend Classes, Inline Functions, Defining Inline Functions Within a Class Parameterized Constructors, Constructors with One Parameter: A Special Case Static Class Members ,Static Data Members ,Static Member Functions ,When Constructors and Destructors Are Executed ,The Scope Resolution Operator, Nested Classes, Local Classes, Passing Objects to Functions ,Returning Objects ,Object Assignment.	
Module -2	**10 Hours**
Arrays, Pointers, References, and the Dynamic Allocation Operators: Arrays of Objects, Creating Initialized vs. Uninitialized Arrays , Pointers to Objects, Type Checking C++ Pointers , The this Pointer, Pointers to Derived Types , Pointers to Class Members , References, Reference Parameters, Passing References to Objects, Returning References, Independent References, References to Derived Types, Restrictions to References, A Matter of Style , C++'s Dynamic Allocation Operators ,Initializing Allocated Memory ,Allocating Arrays ,Allocating Objects ,The nothrow Alternative ,The Placement Form of new. **Function Overloading, Copy Constructors and Default Arguments:** Function Overloading, Overloading Constructors, Overloading a Constructor to Gain Flexibility, Allowing Both Initialized and	

Uninitialized Objects Copy Constructors , Finding the Address of an Overloaded Function , The overload Anachronism, Default Function Arguments, Default Arguments vs. Overloading, Function Overloading and Ambiguity.	
Module -3	**10 Hours**
Operator Overloading: Creating a Member Operator Function, Creating Prefix and Postfix Forms, of the Increment and decrement Operators , Overloading the Shorthand Operators, Operator Overloading Restrictions, Operator Overloading Using a Friend Function, Using a Friend to Overload ++ or – –, Friend Operator Functions Add Flexibility , Overloading new and delete, Overloading new and delete for Arrays, Overloading the nothrow Version of new and delete, Overloading Some Special Operators, Overloading [ ] , Overloading ( ), Overloading –>, Overloading the Comma Operator. **Inheritances:** Base-Class Access Control, Inheritance and protected Members, Protected Base-Class Inheritance, Inheriting Multiple Base Classes, Constructors, Destructors, and Inheritance, When Constructors and Destructors Are Executed , Passing Parameters to Base-Class Constructors, Granting Access, Virtual Base Classes.	
Module -4	**10 Hours**
Virtual Functions and Polymorphism: Virtual Functions, Calling a Virtual Function Through a Base, Class Reference, The Virtual Attribute Is Inherited, Virtual Functions Are Hierarchical, Pure Virtual Functions, Abstract Classes, Using Virtual Functions, Early vs. Late Binding. **Templates:** Generic Functions, A Function with Two Generic Types, Explicitly Overloading a Generic Function, Overloading a Function Template, Using Standard Parameters with Template Functions , Generic Function Restrictions , Applying Generic Functions, A Generic Sort, Compacting an Array, Generic Classes, An Example with Two Generic Data Types, Applying Template Classes: A Generic Array Class, Using Non-Type Arguments with Generic Classes, Using Default Arguments with Template Classes, Explicit Class Specializations, The typename and export Keywords, The Power of Templates .**Exception Handling:** Exception Handling Fundamentals, Catching Class Types, Using Multiple catch Statements, Handling Derived-Class Exceptions, Exception Handling Options, Catching All Exceptions, Restricting Exceptions, Rethrowing an Exception, Understanding terminate( ) and unexpected( ), Setting the Terminate and Unexpected Handlers, The uncaught_exception( ) Function, The exception and bad_exception Classes Applying Exception Handling.	
Module -5	**10 Hours**
Standard C++ I/O Classes : Old vs. Modern C++ I/O, C++ Streams, The C++ Stream Classes, C++'s Predefined Streams, Formatted I/O, Formatting Using the ios Members, Setting the Format Flags, Clearing Format Flags, Overloading << and >>, Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions **C++ File I/O:** fstream> and the File Classes, Opening and Closing a File, Reading and Writing Text Files, Unformatted and Binary I/O, Characters vs. Bytes, put( ) and get( ), read( ) and write( ), More get( ) Functions, getline( ) , Detecting EOF, The ignore( ) Function, peek( ) and putback( ), flush( ), Random Access, Obtaining the Current File Position , I/O Status, Customized I/O and Files. **STL** : An Overview of the STL, Containers, Algorithms, Iterators and Function Objects, The String Class, Vector Class, Lists, Maps.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Book

1. Herbert Schildt: C++ The Complete Reference, 4th Edition, Tata McGraw Hill, 2014. (Listed topics only from Chapters 11,12,13,14, 15, 16, 17, 18, 19, 20, and21)
2. K R Venugopal, RajkumarBuyya,TRavishanker: Mastering C++,Tata McGraw Hill.

Reference Books:

1. Stanley B. Lippmann, JoseeLajore: C++Primer, 4th Edition, Addison Wesley.
2. Stephen Prata : C++ Primer Plus, 6th Edition, Pearson Education.
3. Object oriented programming with C++, E. Balaguruswamy, Tata McGraw Hill.

DATABASE MANAGEMENT SYSTEM

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – II

Subject Code	16MCA23	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

- **CO1:** Demonstrate the fundamentals of data models and conceptualize and depict a database system and Make use of ER diagram in developing ER Model
- **CO2:** To Summarize the SQL and relational database design.
- **CO3:** Illustrate transaction processing, concurrency control techniques and recovery
- **CO4:** Inference the database design in the real world entities.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Introduction: Characteristics of Database approach, Actors on the Scene, Workers behind the scene, Advantages of using DBMS approach, Data models, schemas and instances, Three -schema architecture and data independence, Database languages and interfaces, the database system environment, Centralized and client -server architectures, Classification of Database Management systems, Entity-Relationship Model: Conceptual Database using high level conceptual data models for Database Design, A Sample Database Application, Entity types, Entity sets Attributes and Keys Relationship types, Relationship Sets, Roles and Structural Constraints Weak Entity Types.	
Module -2	**10 Hours**
Relational Model Relational Model and Relational Algebra: Relational Model Concepts, Relational Model Concepts, Relational Model Constraints and Relational Database Schema Update Operations, Transactions and Dealing with Constraint violations, Unary Relational operations, Relational Algebra Operations from Set Theory, Binary Relational Operations, JOIN and DIVISION, Additional Relational Operations, Examples of Queries in Relational Algebra Relational Database Design Using ER-to-Relational Mapping	
Module -3	**10 Hours**
Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic structure of SQL Queries, Additional Basic Operations, Null values, Aggregate Functions, nested Sub queries, Modification of the Database, Join Expressions, Views, Transactions, Integrity Constraints, SQL Data Types and Schemas, Authorization. Database programming issues and techniques, Embedded SQL.	
Module -4	**10 Hours**
Database Design: Informal Design Guidelines for Relation Schemas, Functional Dependencies, Normal Forms based on Primary Keys, General Definitions of 2nd and 3rd Normal Forms,	

Boyce Codd Normal Forms, Stored Procedures and functions,Triggers.	
Module -5	**10 Hours**
Transaction Management: Transaction Concept, A Simple Transaction Model, Transaction Atomicity and Durability, Serializability, Transaction Isolation and Atomicity, Transaction Isolation Levels, Implementation of Isolation Levels. Concurrency Control: Lock Based Protocols, Deadlock Handling. Recovery System: Failure Classification, Storage, Recovery and Atomicity, Recovery Algorithm.	
Question paper pattern: • The question paper will have ten questions. • Each full question consists of 16 marks. • There will be 2full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Addison -Wesley, 2011. 2. Silberschatz, Korth and Sudharshan Data base System Concepts,6th Edition, Tata McGraw Hill, 2011.	
Reference Books: 1 C.J. Date, A. Kannan, S. Swamynatham: An Introduction to Database Systems, 8th Edition, Pearson education,2009. 2 Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw-Hill, 2003.	

Operating Systems

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – II

Subject Code	16MCA24	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand the Basics of Computer and Operating Systems Structure

CO2: Realize the concept of Process Management and Mutual Execution

CO3: Understand the concepts of the Deadlock and different approaches to memory management.

CO4: Learn the concepts of file system

CO5: Understand the concepts of Computer Security.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Computer and Operating Systems Structure BasicElements,ProcessorRegisters,Instruction Execution, The Memory Hierarchy,CacheMemory,I/OCommunicationTechniques,IntroductiontoOperatingSystem,MainframeSystems, DesktopSystems,MultiprocessorSystems,DistributedSystems,ClusteredSystems,Real-TimeSystems,HandheldSystems,FeatureMigration,ComputingEnvironments. **SystemStructures:** SystemComponents,Operating–SystemServices,SystemCalls,SystemPrograms,SystemStructure,VirtualMachines,SystemDesign andImplementation,SystemGeneration	
Module -2	**10 Hours**
Process Management and Mutual Execution Process,ProcessStates,ProcessDescription,ProcessControl, Executionof theOperatingSystem,SecurityIssues,Processes and Threads,Symmetric Multiprocessing(SMP),Microkernels, CPU Scheduler and Scheduling. Principles of Concurrency, Mutual Exclusion: Hardware Support, Semaphores , Monitors , Message Passing, Readers/Writes Problem.	
Module -3	**10 Hours**
Deadlock and Memory Management 1 0Hours PrinciplesofDeadlock,DeadlockPrevention,Deadlock Avoidance, DeadlockDetection,AnIntegratedDeadlockStrategy,DiningPhilosophersProblem MemoryManagement: Swapping,ContiguousMemoryAllocation,Paging,Segmentation,SegmentationwithPaging,Dem	

andPaging,ProcessCreation,PageReplacement,AllocationofFrames,Thrashing	
Module -4	**10 Hours**
File System and Secondary Storage FileConcept,AccessMethods,Directory Structure,FileSystem Mounting,FileSharing,Protection,File–SystemStructure,File–SystemImplementation,DirectoryImplementation,AllocationMethods, Free–SpaceManagement, Disk Structure,DiskScheduling,DiskManagement.	
Module -5	**10 Hours**
ComputerSecurity and CasestudyofLinuxOperatingsystem TheSecurityProblem,UserAuthentication,ProgramThreats,SystemThreats. Linux System Linuxhistory,DesignPrinciples,Kernelmodules,Process ,management,scheduling, Memory management, File systems, Input and output,Inter-process communications	
Question paper pattern: • The question paper will have ten questions. • Each full question consists of 16 marks. • There will be 2full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. Silberschatz, Galvin, Gagne, "Operating System Concepts" John Wiley, Sixth Edition, 2004 2. William Stallings, "Operating System Internals and Design Principles" Pearson, 6th edition, 2012	
Reference Books: 1. Chakraborty,"OperatingSystems"JaicoPublishingHouse,2011. 2. DhananjayM.Dhamdhere,"OperatingSystems–AConcept–BasedApproach",TataMcGraw–Hill,3rd Edition, 2012.	

SYSTEM SOFTWARE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – II

Subject Code	16MCA25	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand the introductory concepts of system software, SIC and SIC/XE machine architecture.

CO2: Understand the design and implementation of Assemblers with implementation examples.

CO3: Design and implement the linkers and loaders, macro processors and respective implementation examples.

CO4: Learn the basic design and working of compilers.

Modules	Teaching Hours
Module -1	**10 Hours**
Machine Architecture & Introduction to Assemblers Introduction, System Software and Machine Architecture, Simplified Instructional Computer (SIC) – SIC Machine Architecture, SIC/XE Machine Architecture, SIC Programming Examples. VAX Architecture, UltraSPARC Architecture. Basic Assembler Function - A Simple SIC Assembler, Assembler Algorithm and Data Structures.	
Module -2	**10 Hours**
Assemblers Machine Dependent Assembler Features - Instruction Formats & Addressing Modes, Program Relocation. Machine Independent Assembler Features – Literals, Symbol-Definition Statements, Expression, Program Blocks, Control Sections and Programming Linking, Assembler Design Operations - One-Pass Assembler, Multi-Pass Assembler, Implementation Examples – MASM Assembler, SPARC Assembler	
Module -3	**10 Hours**
Loaders and Linkers Basic Loader Functions - Design of an Absolute Loader, A Simple Bootstrap Loader, Machine-Dependent Loader Features – Relocation, Program Linking, Algorithm and Data Structures for a Linking Loader; Machine-Independent Loader Features – Automatic Library Search, Loader Options, Loader Design Options - Linkage Editor, Dynamic Linkage, Bootstrap Loaders, Implementation Examples – MS-DOS linker, SunOS Linker, Cray MPP linker	
Module -4	**10 Hours**

Macro Processor Basic Macro Processor Functions - Macro Definitions and Expansion, Macro Processor Algorithm and Data Structures, Machine-Independent Macro Processor Features –Concatenation of Macro Parameters, Generation of Unique Labels, Conditional Macro Expansion, Keyword Macro Parameters, Macro Processor Design Options – Recursive Macro Expansion, General- Purpose Macro Processors, Macro Processing Within Language Translators, Implementation Examples - MASM Macro Processor, ANSI C Macro Processor., ELENA macro processor	
Module -5	**10 Hours**
Compilers Basic Compilers Functions- Grammars, Lexical Analysis, Syntactic Analysis, Code Generation. Machine Dependent Compiler Features- Intermediate Form of the Program, Machine dependent code Optimization. Machine Independent Compiler Features- Structured variables, Machine Independent code Optimization. Compiler Design Options- Division into passes, Interpreters, P-code Compilers, Compiler-Compilers, SunOS C compiler, YACC Compiler-compiler.	
Question paper pattern: • The question paper will have ten questions. • Each full question consists of 16 marks. • There will be 2full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. Leland.L.Beck: System Software, 3rd Edition, Addison-Wesley, 1997.	
Reference Books: 1. J.Nithyashri, “System Software”, 2nd Edition, Tata McGraw Hill, 2010	

PYTHON PROGRAMMING LABORATORY			
Laboratory Code	16MCA26	CIE Marks	20
Number of Lecture Hours/Week	01Hour Tutorial/ Instructions 02 Hours Laboratory	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03
CREDITS – 02			
Course Outcome (CO): At the end of this course, the students will be able to **CO1:** Apply object-oriented programming concepts to develop dynamic interactive Python applications. **CO2:** Use the procedural statements: assignments, conditional statements, loops, method calls and arrays **CO3:** Design, code, and test small Python programs with a basic understanding of top-down design. **CO4:** Learn how to create GUI andsolve real-world problem using language idioms, data structures and standard library			
Laboratory Experiments:			
1.Write a program to sum all the elements from n1 to n2 where n1 and n2 are positive integers			
2. Input an array of n numbers and find separately the sum of positive numbers and negative numbers.			
3. Write a program to search an element using linear search			
4. Write a program to search an element using binary search.			
5. Write a program to simulate stack.			
6. Using a stack evaluate an arithmetic expression.			
7. Write a program to multiply two matrices.			
8. Write a program to find the roots of a quadratic equation			
9.Write a program to Insert a number in a sorted array.			
10. Write a Python Program to check whether the given string is palindrome or not using built in string manipulation methods.			
11. Write a Python Program to read a word and prints the number of letters, vowels and percentage of vowels in the word using dictionary			
12. Write a Python Program to check a given sentence is a pangram or not using function/Module.			
13.Write a Python Event driven Program for file operations Press 1: to open file in read mode 2: open the file in write mode 3: current position of the file pointer #4: Reposition the pointer at the beginning 5: exit.			
14.Write an Object oriented Python program to create two Time objects: currentTime, which contains the current time; and breadTime, which contains the amount of time it takes for a bread maker to make bread. Then we'll use addTime to figure out when the bread will be done. Write the printTime function to display the time when the bread will be done by the bread maker.			

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfroma lot of all14 questions.

Note 2: Change of program is not permitted in the Practical Examination.

OBJECT ORIENTED PROGRAMMING USING C++ LABORATORY

Laboratory Code	16MCA27	CIE Marks	20
Number of Lecture Hours/Week	01 Hours Tutorial/ Instructions 02 Hours Laboratory	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03

CREDITS – 02

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Apply and implement major programming and object oriented concepts like function overloading, operator overloading, Encapsulations, and inheritance, message passing to solve real-world problems.

CO2: Use major C++ features such as Virtual functions, Templates for data type independent designs and File I/O to deal with large data sets.

CO3: Analyze, design and develop solutions to real-world problems applying OOP Concepts of C++.

Laboratory Experiments:

PART-A

1. Write a C++ program to find the sum for the given variables using function with default arguments.
2. Write a C++ program to swap the values of two variables and demonstrates a function using call by value.
3. Write a C++ program the swap the values of two variables and demonstrates a function using Call by reference.
4. Write a program to find the largest, smallest & second largest of three numbers. (use inline function MAX and MIN to find largest & smallest of 2 numbers)
5. Write a program to calculate the volume of different geometric shapes like cube, cylinder and sphere and hence implement the concept of Function Overloading.
6. Write a C++ program to create a template function for Bubble Sort and demonstrate sorting of integers and doubles.

PART-B

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. Friend functions and friend classes:

   a) Write a program to define class name HUSBAND and WIFE that holds the income respectively. Calculate and display the total income of a family using Friend function.

   b) Write a program to accept the student detail such as name and 3 different marks by get_data() method and display the name and average of marks using display() method. Define a friend class for calculating the average of marks using the method mark_avg().

4. 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.

5. 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.

6. Write a program to create an HUMAN class with features Head, Legs, Hands.(NOTE: Head, Legs and Hands are of integer/float types)
   a. Create an object HUMAN1 using default constructor. (Default features to have 1 Head, 2 Legs and 2 Hands)
   b. Create an object HUMAN2 with customized inputs using Parameterized Constructor
   C. Create an object HUMAN3 using existing object HUMAN1 (Copy Constructor).
   D. Create an object HUMAN4 using Default Arguments Constructor (1 Head, 2 Legs and 2 Hands.
   E. All Humans die after their lifetime.(Destructor)

7. Demonstrate Simple Inheritance concept by creating a base class FATHER with data members FirstName, SurName, DOB and BankBalance and creating a derived class SON, which inherits SurName and BankBalance feature from base class but provides its own feature FirstName and DOB. Create and initialize F1 and S1 objects with appropriate constructors and display the Father & Son details.

8. 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.

9. I/O streams and functions.
   Write a program to implement FILE I/O operations on characters. I/O operations includes inputting a string, Calculating length of the string, Storing the string in a file, fetching the stored characters from it, etc.

10. Write a program to implement Exception Handling with minimum 5 exceptions Classes including two built-in exceptions.

11. Write a program to concatenate 2 strings using STL String class functions.

12. Write a simple C++ program to store and display integer elements using STL Vector class.

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfromPART-A and PART-Beach.

Note 2: Change of program is not permitted in the Practical Examination.

DATABASE MANAGEMENT SYSTEMS LABORATORY			
Laboratory Code	16MCA28	CIE Marks	20
Number of Lecture Hours/Week	01Hour Tutorial/ Instructions 02 Hours Laboratory	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03
CREDITS – 02			

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand, appreciate the underlying concepts of database technologies

CO2: Able to create database with different types of integrity constraints and use the SQL commands such as DDL, DML, DCL, TCL to access data from database objects.

CO3: Design and implement a database schema for a given problem domain

CO4: Perform embedded and nested queries

CO5: Take up real world problems independently

DBMS Lab Experiments:

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

1 Create the following tables with properly specifying Primary keys, Foreign keys and solve the following queries.

BRANCH(**Branchid**,Branchname,HOD)
STUDENT(**USN**,Name,Address,Branchid,sem)
BOOK(Bookid,Bookname,Authorid,Publisher,Branchid)
AUTHOR(Authorid,Authorname,Country,age)
BORROW(USN,Bookid,Borrowed_Date)
Queries:

1 List the details of Students who are all Studying in 2ndsem MCA.
2 List the students who are not borrowed any books.
3 Display the USN, Student name, Branch_name, Book_name, Author_name ,Books_Borrowed_Date of 2ndsem MCA Students who borrowed books.
4 Display the number of books written by each Author.
5 Display the student details who borrowed more than two books.

6 Display the student details who borrowed books of more than one Author.
7 Display the Book names in descending order of their names.
8 List the details of students who borrowed the books which are all published by the same Publisher.

2 **Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the following queries.**

Consider a Cricket Tournament “ABC CUP” organized by an organization. In the tournament there are many teams are contesting each having a Teamid,Team_Name, City, a coach. Each team is uniquely identified by using Teamid. A team can have many Players and a captain. Each player is uniquely identified by Playerid, having a Name, and multiple phone numbers,age. A player represents only one team. There are many Stadiums to conduct matches. Each stadium is identified using Stadiumid, having a stadium_name,Address (involves city,area_name,pincode).A team can play many matches. Each match played between the two teams in the scheduled date and time in the predefined Stadium. Each match is identified uniquely by using Matchid. Each match won by any of the one team that also wants to record in the database. For each match man_of_the match award given to a player.

Queries:

1 Display the youngest player (in terms of age) Name, Team name , age in which he belongs of the tournament.
2 List the details of the stadium where the maximum number of matches were played.
3 List the details of the player who is not a captain but got the man_of _match award at least in two matches.
4 Display the Team details who won the maximum matches.
5 Display the team name where all its won matches played in the same stadium.

3 **Consider the following Scenario and design an ER-Diagram, map the designed ER-diagram into a Relational model.**

Consider an organization “ABC” having many employees. An employee works for one department. Each employee identified by using Empid, having Name, address (described as House_no, city, district, state, pin code) and more than one phone numbers. Department identified by using Dno, having Dname, Dlocation. Each Department having a manager . Each department having many employees. There are many Projects , each project is controlled by the department. Each Project uniquely identified by Pno, having Project_name,Project_location. An employee works on many Projects. Number of hours per week worked on each project by an Employee also needs to be recorded in the database . A project is worked by many employees. Each employee supervised by the supervisor. Employee having many dependents. Dependents having thedependent_name, gender, age, address. Dependents are identified by Empid.

T1(Empid, Emp_Name,city, district, state, pin_code, phoneno, Dno,Dname,Dlocation, Dept_mgr_id, Pno, Project_name, Project_location, Number_of_Hours,Supervisor_Empid, Dependent_name, gender, address)

Deduce the above Relation T1 into the 3NF and then solve the following queries.

Queries:

1. Display the details of the employees who are working on both the projects havingproject_no 5 and 10.
2. Display the details of employees having atleast two dependents.
3. Display the project name on which more number of employees are working.
4. Retrieve the employees who do not have any dependents.
5. Display the Employee details whose total number of hours per week working on various projects is maximum than all other employees.
6. create a view to display the number of employees working in each department.

4 **Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries.**

A country can have many Tourist places . Each Tourist place is identified by using tourist_place_id, having a name, belongs to a state, Number of kilometers away from the capital city of that state,history. There are many Tourists visits tourist places every year. Each tourist is identified uniquely by using Tourist_id, having a Name, age, Country and multiple emailids. A tourist visits many Tourist places, it is also required to record the visted_date in the database. A tourist can visit a Tourist place many times at different dates. A Tourist place can be visited by many tourists either in the same date or at different dates.

Queries:

1 List the state name which is having maximum number of tourist places.
2 List details of Tourist place where maximum number of tourists visited.
3 List the details of tourists visited all tourist places of the state “KARNATAKA”.
4 Display the details of the tourists visited at least one tourist place of the state, but visited all states tourist places.
5 Display the details of the tourist place visited by the tourists of all country.

5 **Design an ER-diagram for the following scenario, Convert the same into a relational model, normalize Relations into a suitable Normal form and then solve the following queries.**

A country wants to conduct an election for the parliament. A country having many constituencies. Each constituency is identified uniquely by Constituency_id, having the Name, belongs to a state,Number_of_voters. A constituency can have many voters. Each voter is uniquely identified by using Voter_id, having the Name, age, address (involves Houseno,city,state,pincode). Each voter belongs to only one constituency. There are many candidates contesting in the election. Each candidates are uniquely identified by using candidate_id, having Name, phone_no, age, state. A candidate belongs to only one party.Thereare many parties. Each party is uniquely identified by using Party_id, having Party_Name,Party_symbol. A candidate can contest from many constituencies under a same party. A party can have many candidates contesting from different constituencies. No constituency having the candidates from the same party. A constituency can have many contesting candidates belongs to different parties. Each voter votes only one candidate of his/her constituencty.

Queries:

1 List the details of the candidates who are contesting from more than one constituencies which are belongs to different states.

2 Display the state name having maximum number of constituencies.

3 Create a stored procedure to insert the tuple into the voter table by checking the voter age. If voter's age is at least 18 years old, then insert the tuple into the voter else display the "Not an eligible voter msg" .

4 Create a stored procedure to display the number_of_voters in the specified constituency. Where the constituency name is passed as an argument to the stored procedure.

5 Create a TRIGGER to UPDATE the count of " Number_of_voters" of the respective constituency in "CONSTITUENCY" table , AFTER inserting a tuple into the "VOTERS" table.

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfroma lot of all the 5 questions.

Note 2: Change of program is not permitted in the Practical Examination.

Computer Networks

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Subject Code	16MCA31	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): The students will be able to

- **CO1**: Understand the types of Networks & Communication medias.
- **CO2**: Identify the components required to build different types of networks
- **CO3**: Understand the functionalities needed for data communication into layers
- **CO4**: Choose the required functionality at each layer for given application
- **CO5**: Understand the working principles of various application protocols

Modules	**Teaching Hours**
Module -1	**10 Hours**
Introduction to Computer Networks and Physical Layer Networking Devices, Classification of Computer Networks, Network Protocol Stack (TCP/IP and ISO-OSI),Network Standardization and Examples of Networks.Data Transmission Concepts, Analog and Digital Data Transmission, Communication media, Digital modulation techniques (FDMA,TDMA,CDMA).	
Module -2	**10 Hours**
Data Link layer Data link layer design issues, Error Detection and Correction Codes, Data Link Protocols (Simplex Stop-and-wait protocol for Error free and noisy channel) and Sliding window protocols. **Medium Access Sub Layer** The Channel Allocation Problem, Multiple access protocols and Examples, Wireless LAN, Bluetooth.	
Module -3	**10 Hours**
Network Layer Network Layer Design issues, Routing algorithms, Congestion Control Algorithms, Quality of Service, Internetworking and The Network Layer in the Internet.	
Module -4	**10 Hours**
The Transport Layer	

The Transport Service, Elements of Transport Protocols, Congestion Control, The Internet Transport Protocol: UDP, The Internet Transport Protocols – TCP.	
Module -5	**10 Hours**
The application Layer DNS: Domain Name Space, Domain Resource Records, Domain Name Servers. Electronic mail: SMTP, The World Wide Web: Static and dynamic web pages, web applications, HTTP, mobile web. Streaming audio and Video: Digital audio and video, streaming stored and live media, Content delivery: Content and internet traffic, content delivery networks, peer-to-peer networks.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

1. “Computer Networks” by Andrew S Tanenbaum, David J Wetheral, 5th Edition, Pearson 2012
   Chapter 1, 2.2, 2.3, 2.5, 2.7, 3.1, 3.2, 3.3, 3.4,4.1, 4.2 (4.2.1-4.2.3), 4.4.1, 4.6 (4.6.1-4.6.2).
   Chapter 5.1, 5.2 (5.2.2, 5.2.4, 5.2.5),5.3.1-5.3.3, 5.4.1-5.4.3, 5.5.1-5.5.3, 5.6.1-5.6.3.

Chapter 6.1-6.5.

Chapter 7.1-7.3, 7.4.1-7.4.4, 7.5.1, 7.5.3, 7.5.4.

2 “Data and Computer Communications” by William Stallings , Above 7th Edition , 2004 Chapter 3

Reference Books:

1. “Computer Networks” Principles, Technologies and Protocols for Network Design, by NATALA OLIFER and VICTOR OLIFER , Wiley, 2010
2. http://www.ietf.org/rfc.html 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

Java Programming

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Subject Code	16MCA32	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand the basic programming constructs of Java. Apply suitable OOP concepts to develop Java programs for a given scenario.

CO2: Illustrate the concepts of Generalization and run time polymorphism applications

CO3: Exemplify the usage of Packages, Interfaces, Exceptions and Multithreading

CO4: Demonstrate Enumerations, Wrappers, Auto boxing, Generics,collection framework and I/O operations

CO5: Implement the concepts of Networking using Java network classes

Modules	Teaching Hours
Module -1	**10 Hours**
Java Programming Fundamentals The Java Language, The Key Attributes of Object-Oriented Programming, TheJava Development Kit, A First Simple Program, Handling Syntax Errors, The Java Keywords, Identifies in Java, The Java Class Libraries. **Introducing Data Types and Operators** 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** 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** 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** 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** 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.	
Module -2	**10 Hours**
A Closer Look at Methods and Classes 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** 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.	
Module -3	**10 Hours**
Interfaces 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** Package Fundamentals, Packages and Member Access, Importing Packages, Static Import **Exception Handling** 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.	
Module -4	**10 Hours**
Multithreaded Programming 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** Enumerations, Java Enumeration are class types, The Values() and Valueof() Methods, Constructors, methods, instance variables and enumerations, Auto boxing, Annotations (metadata)	
Module -5	**10 Hours**
Applets 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. **Networking with Java.net** Networking fundamentals, The Networking classes and Interfaces, The InetAddress class, The Socket Class, The URL class, The URLConnection Class, The HttpURL Connection Class. **Exploring Collection Framework** Collections Overview, The Collection Interfaces, The collection Classes. The Arrays Class.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module

Text Books:

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

Reference Books:

1. Java Programming by Hari Mohan Pandey, Pearson Education, 2012.
2. Java 6 Programming, Black Book, KoGenT ,Dreamtech Press, 2012.
3. Java 2 Essentials, Cay Hortsmann, second edition, Wiley

Analysis and Design of Algorithms

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Subject Code	16MCA33	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): ***At the end of this course, the students will be able to***

CO1: Categorize problems based on their characteristics and practical importance.

CO2: Develop Algorithms using iterative/recursive approach

CO3: Compute the efficiency of algorithms in terms of asymptotic notations

CO4: Design algorithm using an appropriate design paradigm for solving a given problem

CO5: Classify problems as P, NP or NP Complete

CO6: Implement algorithms using various design strategies and determine their order ofgrowth.

Modules	Teaching Hours
Module -1	**10 Hours**
Introduction, Fundamentals of the Analysis of Algorithm Efficiency Notion of Algorithm, Fundamentals of Algorithmic Problem Solving, Important Problem Types, Fundamental data Structures. Analysis Framework, Asymptotic Notations and Basic efficiency classes, Mathematical analysis of Recursive and Non-recursive algorithms.	
Module -2	**10 Hours**
Brute Force: Selection Sort and Bubble Sort, Sequential Search, Exhaustive search and String Matching. **Divide-and-Conquer** Mergesort, Quicksort, Binary Search, Binary tree Traversals and related properties, Multiplication of large integers.	
Module -3	**10 Hours**
Decrease-and-Conquer Insertion Sort, Depth First and Breadth First Search, Topological sorting, Algorithms for Generating Combinatorial Objects: generating permutations. **Greedy Technique** Prim's Algorithm, Kruskal's Algorithm, Dijkstra's Algorithm, Huffmann Trees.	
Module -4	**10 Hours**
Space and Time Tradeoffs Sorting by Counting, Input Enhancement in String Matching, Hashing. **Dynamic Programming** Computing a binomial coefficient, Warshall's and Floyd's Algorithms, The Knapsack Problem and Memory Functions	
Module -5	**10 Hours**
Limitations of Algorithm Power Lower-Bound Arguments, Decision Trees, P, NP and NP-Complete Problems. **Coping with Limitations of Algorithm Power** Backtracking: n-Queens problem, Hamiltonian Circuit Problem, Subset – Sum Problem. Branch-and-Bound: Assignment Problem, Knapsack Problem, Traveling Salesperson Problem.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. AnanyLevitin: Introduction to the Design and Analysis of Algorithms, Pearson Education, 2nd Edition.(Chapters 1.1-1.4, 2.1-2.4, 3.1, 3.2, 3.4, 4.1-4.5, 5.1-5.4, 7.1-7.3, 8.1, 8.2, 8.4, 9.1-9.4, 11.1-11.3, 12.1-12.2)

Reference Books:

1. Coremen T.H., Leiserson C.E., and Rivest R.L.: Introduction to Algorithms, PHI 1998.
2. Horowitz E., Sahani S., Rajasekharan S.: Computer Algorithms, Galgotia Publication 2001.
3. Michael T Goodrich and Roberto Tamassia : Algorithm Design, Wiley India
4. R C T Lee, S S Tseng, R C Chang, Y T Tsai : Introduction to Design and Analysis of Algorithms: A Strategic Approach, Tata McGraw Hill

Software Engineering

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Subject Code	16MCA34	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

- **CO1**: Categorize problems based on their characteristics and practical importance.
- **CO2**: Apply the correct process models for software development.
- **CO3**: Apply the techniques, skills, and modern engineering tools necessary for engineering practice.
- **CO4**: Define, formulate and analyze a problem as per the testing techniques.
- **CO5**: Apply new Generation of Software Engineering Technology to Meet Current and Future Industrial Challenges of Emerging Software Trends.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Overview Introduction: Professional Software Development Attributes of good software, software engineering diversity, IEEE/ ACM code of software engineering ethics, case studies **Software Process & Agile Software Development** Software Process models: waterfall, incremental development, reuses oriented, Process activities; Coping with change, The rational Unified process. Agile methods, Plan-driven and agile Development, Extreme Programming, Agile project management, Scaling agile methods.	
Module -2	
Requirements Engineering Functional and non-functional requirements, The software requirements document, Requirements specification, Requirements engineering processes, Requirement elicitation and analysis, Requirements validation, Requirements management **Component-based software engineering** Components and component model, CBSE process, Component composition.	**10 Hours**
Module -3	**10 Hours**
System Modeling, Architectural Design 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	
Module -4	**10 Hours**
Design and implementation Design: Design concepts, Function oriented design, detailed design, verification, matrix (Complexity matrix for function oriented design) **Distributed Software engineering** Distributed system issues, Client-server computing, Architectural patterns for distributed systems, Software as a service.	

Module -5	**10 Hours**
Planning a software Project Process planning, Effort estimation, Project scheduling and staffing, Software configuration management plan, Quality plan, Risk Management, Project monitoring plan. **Software Testing** Testing fundamentals, Black-box testing, White-box testing, Testing process	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Ian Sommerville : Software Engineering, 9th edition, Person Education Ltd, 2011. (Chapters-: 1, 2, 3, 4, 5, 17, 18)
2. PankajJalote: Software Engineering, Wiley India Pvt Ltd (2010) (Chapters-:4, 6.1, 6.2, 6.5, 6.6)

Reference Books:

1. Roger S Pressman: Software Engineering-A Practitioners approach, 6th Edition, McGraw-Hill, 2010.
2. Hans Van Vliet: Software Engineering Principles and Practices, 3rd Edition, Wiley–India, 2010

Computer Networks Laboratory

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Laboratory Code	16MCA36	CIE Marks	20
Number of LAB Hours/Week	01Hr Tutorial/Instructions 02 Hours Laboratory	SEE Marks	80
		SEE Hours	03

CREDITS – 02

Course Outcome (CO): ***At the end of this course, the students will be able to***

CO1: Understandthe basic terminologies used for computer networking.

CO2: Understand the functions of layers in the Internet Model.

CO3: Demonstrate application layer protocols used for process to process communication.

CO4: Demonstrate subnetting and routing mechanisms for a given network topology. Exemplify link layer functionalities.

CO5: Describe the components and working of wireless networks.

PART – A

Note: Implement the following Computer Networks concepts using C/C++

1. Write a program for distance vector algorithm to find suitable path for transmission.

2. Using TCP/IP sockets, write a client-server program to make the client send the file name and to make the server send back the contents of the requested file if present.

3. Implement the above program using message queues or FIFOs as IPC channels.

4. Write a program for Hamming code generation for error detection and correction.

5. Write a program for congestion control using leaky bucket algorithm.

PART – B

Note: Simulate the following Computer Networks concepts using any network simulators.

1.Simulate a three nodes point — to — point network with duplex links between them. Set the queue size and vary the bandwidth and find the number of packets dropped.

2. Simulate the 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.

3.Simulate to study transmission of packets over Ethernet LAN and determine the number of packets drop destination.

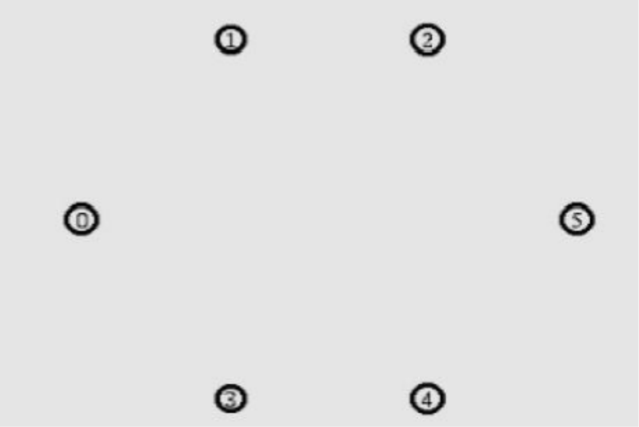
4. Write a TCL Script to simulate working of multicasting routing protocol and analyze the throughput of the network.

5. Simulate the different types of internet traffic such as FTP and TELNET over a wired network and

analyze the packet drop and packet delivery ratio in the network.

6. Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

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.

Note 1: InthepracticalExaminationstudent has to executeone programfromPART-A and one from PART-B.

Note 2: Change of program is not permitted in the Practical Examination.

Java Programming Laboratory

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Laboratory Code	16MCA37	CIE Marks	20
Number of LAB Hours/Week	01Hr Tutorial/ Instructions	SEE Marks	80
	02 Hours Laboratory	SEE Hours	03

CREDITS – 02

Course Outcome (CO): At the end of this course, the students will be able to

CO1: UnderstandJava programming language fundamentals and run time environment.

CO2: Acquire knowledge and skill necessary to write java programs.

CO3: Learn the object oriented concepts and its implementation in Java

CO4: Implement the multithreading and client side programming.

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 to create an enumeration Day of Week with seven values SUNDAY through SATURDAY. Add a method isWorkday( ) 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 Stackoperations 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 1: InthepracticalExaminationstudent has to executeoneprogramfroma lot of all the 14 questions.

Note 2: Change of program is not permitted in the Practical Examination.

Analysis and Design of Algorithms Laboratory

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Laboratory Code	16MCA38	CIE Marks	20
Number of LAB Hours/Week	01Hr Tutorial /Instructions	SEE Marks	80
	02 Hours Laboratory	SEE Hours	03

CREDITS – 02

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Implement the concepts of time and space complexity, divide-and-conquer strategy, dynamic programming, greedy and approximate algorithms.

CO2: Describe the methodologies of how to analyze an algorithm

CO3: Choose a better algorithm to solve the problems

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 Insertion 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.

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 graph.

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

10 Find a subset of a given set S = {s1,s2,.....,sn} 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.

11 a). Implement Horspool algorithm for String Matching.
b). Find the Binomial Co-efficient using Dynamic Programming.

12 Find Minimum Cost Spanning Tree of a given undirected graph using Prim's algorithm.

13 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.

14 Implement N Queen's problem using Back Tracking.

Note 1: InthepracticalExaminationstudent has to executeoneprogramfroma lot of all the 14 questions.

Note 2: Change of program is not permitted in the Practical Examination.

Intellectual Property Rights

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER –III

Subject Code	16MCA351	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Identify which of the four main different types of intellectual property rights may be presented by an output

CO2: Analyze an innovative or creative output in terms of intellectual property rights generated

CO3: Discuss the appropriateness, or not, of registering an intellectual property right

CO4: Apply the appropriate ownership rules to intellectual property he / she has been involved in creating

CO5: Suggest ways of exploiting intellectual property rights created in his / her own work.

Modules	**Teaching Hours**
Module -1	**8 Hours**
Introduction to Intellectual Property Law – The Evolutionary Past - The IPR Tool Kit- Para -Legal Tasks in Intellectual Property Law – Ethical obligations in Para Legal Tasks in Intellectual Property Law - Introduction to Cyber Law – Innovations and Inventions Trade related Intellectual Property Right.	
Module -2	**8 Hours**
Introduction to Trade mark – Trade mark Registration Process – Post registration Procedures – Trade mark maintenance - Transfer of Rights - Inter parties Proceeding – Infringement - Dilution Ownership of Trade mark – Likelihood of confusion - Trademarks claims – Trademarks Litigations – International Trade mark Law	
Module -3	**8 Hours**
Introduction to Copyrights – – Principles of Copyright Principles -The subjects Matter of Copy right – The Rights Afforded by Copyright Law – Copy right Ownership, Transfer and duration – Right to prepare Derivative works – Rights of Distribution – Rights of Perform the work Publicity Copyright Formalities and Registrations - Limitations - Copyright disputes and International Copyright Law – Semiconductor Chip Protection Act	
Module -4	**8 Hours**
The law of patents-patent searches –Patent ownership and transfer-Patent infringement-International Patent Law	
Module -5	**8 Hours**
Introduction to Trade Secret – Maintaining Trade Secret – Physical Security – Employee Limitation - Employee confidentiality agreement - Trade Secret Law - Unfair Competition – Trade Secret Litigation – Breach of Contract – Applying State Law.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1 DebiragE.Bouchoux: “Intellectual Property”. Cengage learning, New Delhi

2. M.Ashok Kumar and Mohd.Iqbal Ali: “Intellectual Property Right” Serials Pub.

3. Cyber Law. Texts & Cases, South-Western’s Special Topics Collections.

Reference Books:

1. PrabhuddhaGanguli: ‘ Intellectual Property Rights” Tata Mc-Graw –Hill, New Delhi
2. J Martin and C Turner “Intellectual Property” CRC PressRichard Stimm “ Intellectual Property” Cengage Learning

ENTERPRISE RESOURCE PLANNING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Subject Code	16MCA352	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcomes(CO):*At the end of this course, the students will be able to*

CO1: Acquire knowledge of
- Benefits of ERP, Process Re-engineering
- Project management and Monitoring

CO2: Analyze the performance of
- Project implementations
- Quality management

CO3: Know how ERP evolves in market place

CO4: Develop the ERP system, ERP with E-Commerce & Internet

Modules	**Teaching Hours**
Module -1	**8 Hours**
Introduction To ERP Overview, Benefits of ERP, ERP and Related Technologies, Business Process Reengineering, Data Warehousing, Data Mining, On–line Analytical Processing, Supply Chain Management	
Module -2	**8 Hours**
ERP Implementation Implementation of Life Cycle, Implementation Methodology, Hidden Costs, Organizing Implementation, Vendors, Consultants and Users, Contracts, Project Management and Monitoring	
Module - 3	**8 Hours**
BusinessModules Business Modules in an ERP Package, Finance, Manufacturing, Human Resource, Plant Maintenance, Materials Management, Quality Management, Sales and Distribution	
Module -4	**8 Hours**
ERPMarket ERP Market Place, SAP AG, PeopleSoft ,Baan Company , JD Edwards World Solutions Company, Oracle Corporation, QAD , System Software Associates.	
Module -5	**8 Hours**
ERP–PresentAndFuture Turbo Charge the ERP System, EIA, ERP and E–Commerce, ERP and Internet, Future Directions in ERP.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Alexis Leon, “ERP Demystified”, Tata McGraw Hill, 1999.
2. Joseph A. Brady, Ellen F. Monk, Bret J. Wangner, “Concepts in Enterprise Resource Planning”, Thomson Learning, 2001.

Reference Books:

1. Vinod Kumar Garg and N.K .Venkata Krishnan, “Enterprise Resource Planning concepts and Planning”, Prentice Hall, 1998.

Jose Antonio Fernandz, “ The SAP R /3 Hand book”, Tata McGraw Hill

MIS & E-Commerce

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –III

Subject Code	16MCA353	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Recognize the roles and uses of technology in business systems, operations and describe organizational structure .

CO2: Equip the students with preliminaries of technologies used in business information systems.

CO3: Familiarize students with the Business applications and E-Commerce initiatives.

Modules	**Teaching Hours**
Module -1	**8 Hours**
Introduction to MIS MIS: Concept, Definition, Role of the Management Information System, Impact of the Management Information System, MIS and User, Management as Control System, A support to the Management, Management effectiveness and Organization Effectiveness. **Strategic Management of Business Performance:** The Concept of Corporate Planning, Essentiality of Strategic Planning, Development of the Business Strategies, Types of Strategies,Tools of Planning, Balance Score Card, Score card and Dash Board.	
Module -2	**8 Hours**
Decision-Making: Decision-Making Concepts, Decision-Making Process, Decision Analysis by Analytical Modeling, Behavioural Concepts in Decision-Making, Organizational Decision-Making, MIS and Decision-Modeling. (Case study) **Information, Knowledge, Business Intelligence:** Information Concepts, Information: A Quality Product, Classification of the Information, Methods of Data and Information Collection, Value of the information, Business intelligence.	
Module -3	**8 Hours**
Technology of Information Systems: Introduction, Data Processing, Transaction Processing, Application Processing, Information System Processing, OLAP(on-line analytical Processing) for Analytical Information, TQM(Total Quality Management) of Information System. Human Factors and User Interface, Evaluation and feasibility of IT Solutions.	
Module -4	**8 Hours**
E-Business Technology: Introduction to E-Business, Models of E-Business:Business to Business(B2B),Business to Customer(B2C), Customer to Business(C2B), Customer to Customer (C2C), Internet and World Wide Web(WWW), Internet/Extranet, Security in E-Business,	

Electronic Payment Systems, Impact of Web on Strategic Management. Enterprise: Content Management System(CMS).	
Module -5	**8 Hours**
E-commerce : Introduction to E-Commerce Technologies: Client-side programming: Important factors in client side, Web page design and production, Overview of HTML, Basic text formatting, Links, Images , Tables, Frames, form server side programming-I: Servlet fundamentals, Server-side Programming II: Database connectivity: Relational database systems, JDBC perspectives and JDBC program example. Business-oriented e-commerce: Feature of B2B e-Commerce, Business models, Integration E-services: Categories of e-services, Web-enabled services, Matchmaking services.	
Question paper pattern: • The question paper will have ten questions. • Each full question consists of 16 marks. • There will be 2 full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. The students will have to answer 5 full questions, selecting one full question from each module	
Text Books: 1. Waman S Jawadekar: Management Information System, 4th Edition, Tata McGraw Hill. (Chapters: 1,3,6,7,16,20) 2. Henry Chan, Raymond Lee, Tharam Dillon, Elizabeth Chang, E Commerce, Fundamentals & Applications, John Wiley & Sons, 2003. Part I & Part II. (chapters: 3 , 4, 5, 12, 13)	
Reference Books: 1 Ralph M Stair and George W Reynolds: Principles of Information Systems, 7th Edition, Thomson, 2010 . 2 Steven Alter: Information Systems - The Foundation of E-Business, 4th Edition, Pearson Education, 2001 3 Rahul De, Managing Information Systems in Business, Government and Society, , Wiley India.	

CYBER SECURITY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – III

Course Code	16MCA354	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course outcomes (CO):*At the end of this course, the students will be able to:*

CO1: Define and illustrate cyber security concepts and applications

CO2: Analyze the working of cyber security principles to system design

CO3: Illustrate appropriate techniques to solve cyber security threats

CO4: Evaluate and implement cyber security through network security protocols

Modules	**Teaching Hours**
Module -1	**08 Hours**
Introduction to Cybercrime and Laws Introduction, Cybercrime: Definition and Origins of the word, Cybercrime andinformation Security, Who are Cybercriminals? Classifications of Cybercrimes.How Criminals Plan Them – Introduction, How Criminals Plan the Attacks, Cybercafeand Cybercrimes, Botnets, Attack Vector, The Indian IT ACT 2000.	
Module -2	**08 Hours**
Tools and Methods used in Cybercrime Introduction, Proxy Server and Anonymizers, Password Cracking, Keyloggers and Spyware,Virus and Warms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQLinjection, Buffer Overflow.	
Module -3	**08 Hours**
Phishing and Identity Theft Introduction, Phishing – Methods of Phishing, Phishing Techniques, Phishing Toolkits andSpy Phishing. Identity Theft – PII, Types of Identity Theft, Techniques of ID Theft. DigitalForensics Science, Need for Computer Cyberforensics and Digital Evidence, DigitalForensics Life Cycle.	
Module -4	**08 Hours**
Unix Command Lines, Backtrack Linux, Mac Ports, Cygwin, Windows Power Shell, NetCatCommands, NetCat Uses, SSH, Data Pipe, Fpipe	
Module -5	**08 Hours**
Network Defense tools Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Packet Characteristic to Filter, Stateless VsStateful Firewalls, NetworkAddress Translation (NAT) and Port Forwarding, the basic of Virtual Private Networks,Linux Firewall, Windows Firewall, Snort: Introduction Detection System.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Anti-Hacker Tool Kit (Indian Edition) by Mike Shema, Publication McGraw Hill. (Chapters: 2, 7, 8, 11)
2. Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives by Nina Godbole and SunitBelpure, Publication Wiley. (Chapters: 1.1, 1.2, 1.3, 1.4, 1.5, 2.1, 2.2, 2.5, 2.6, 2.7, 6.4, 5.2.1, 5.2.2, 5.2.5, 5.3.1, 5.3.2, 5.3.3, 4.2,4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11)

Reference Books:

1. Marjie T. Britz - Computer Forensics and Cyber Crime: An Introduction - Pearson
2. Chwan-Hwa (John) Wu,J. David Irwin - Introduction to Computer Networks and Cyber security - CRC Press
3. Bill Nelson, Amelia Phillips, Christopher Steuart - Guide to Computer Forensics and Investigations - Cengage Learning

ADVANCED JAVA PROGRAMMING
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – IV

Subject Code	16MCA41	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Learn the concept of Servlet and its life cycle
CO2: Understand JSP tags and its services
CO3: Create packages and interfaces
CO4: Build Database connection
CO5: Develop Java Server Pages applications using JSP Tags.
CO6: Develop Enterprise Java Bean Applications

Modules	**Teaching Hours**
Module -1	**10 Hours**
Servlet Servlet Structure, Servlet packaging, HTML building utilities, Lifecycle, SingleThreadModel interface, Handling Client Request: Form Data, Handling Client Request: HTTP Request Headers. Generating server Response: HTTP Status codes, Generating server Response: HTTP Response Headers, Handling Cookies, Session Tracking. Overview of JSP: JSP Technology, Need of JSP, Benefits of JSP, Advantages of JSP, Basic syntax,	
Module -2	**10 Hours**
JSP and Controlling the Structure of generated servlets 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 autoflush attributes, info attribute, errorPage and iserrorPage attributes, isThreadSafe Attribute, extends attribute, language attribute, Including files and applets in jsp Pages, using java beans components in JSP documents	
Module - 3	**10 Hours**
Annotations and Java Beans Creating Packages, Interfaces, JAR files and Annotations. The core java API package, New java.Lang Sub package, Built-in Annotations with examples. 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.	
Module – 4	**10 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: The Problem domain, Breakup responsibilities, Code Smart not hard, the Enterprise java bean specification. Components Types.	
Module -5	**10 Hours**
EJB and Server Side Component Models Server Side Component Types, Session Beans, MessageDriven Beans, Entity Beans, The Java Persistence Model. Container services. Dependency Injection, Concurrency, Instance pooling n caching, Transactions, security, Timers, Naming and object stores, Interoperability, Life Cycle Callbacks, Interceptors, platform integration. Developing your first EJB. Preparation, Definitions, naming conventions, convention for the Examples, coding the EJB, the contract, the bean Implementation class, out of Container Testing, Integration Testing. Models: The Stateless Session Bean, the Stateful Session Bean, the Singleton SessionBean, Message-Driven Beans. EJB and PERSISTENCE. Persistence Entity manager Mapping Persistence objects, Entity Relationships	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Marty Hall, Larry Brown. Core Servlets and Java Server Pages. Volume 1: Core Technologies. 2nd Edition. (Chapter 3,4,5,6,7,8,9,10,11,12,13,14).
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:

1. Michael Sikora, EJB 3 Developer Guide, A practical guide for developers and architects to the Enterprise Java Beans Standard, Shroff Publishers & Distributors PVT LTD. July 2008.
2. Herbert Schildt, Java The Complete Reference, 8th Edition. Comprehensive coverage of the Java Language. Tata McGraw-Hill Edition – 2011.

Advanced Web Programming

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	16MCA42	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcomes (CO):***At the end of this course, the students will be able to:***

CO1: Acquire knowledge of

- Build the Web Applications using JQuery, PHP, Ruby and D3.js.
- Model-View-Controller (MVC) Architecture.

CO2: Design the Web Pages using Ruby, Rails and Layouts.

CO3: Apply the knowledge gained in the Building a web portals.

CO4: Evaluate web site performance against user acceptance testing.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Introduction to jQuery Introducing jQuery, jQuery fundamentals, Creating the wrapped element set, Bringing pages to life with jQuery, Understanding the browser event models, The jQuery Event Model, Sprucing up with animations and effects.	
Module -2	**10 Hours**
Introduction to PHP and Building Web applications with PHP 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, Tracking users, cookies, sessions, Using databases, Handling XML.	
Module -3	**10 Hours**
Introduction to Ruby and Introduction to Rails Origins and uses of Ruby, Scalar types and their operations, Simple input and output, Control statements, Arrays, Hashes, Methods, Classes, Code blocks and iterates, Pattern matching. Overview of Rails, Document requests, Processing forms, Layouts. Rails applications with Databases.	
Module -4	**10 Hours**
Web 2.0 and Web Services 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. Web Services: SOAP, RPC Style SOAP, Document style SOAP. WSDL, REST services, JSON format, What is JSON? Array literals, Object literals, Mixing literals, JSON Syntax, JSON Encoding and Decoding, JSON versus XML.	

Module -5	**10 Hours**
D3.js (Data Driven Documents): Data visualization tool for web apps Introduction to D3 : Building a Simple Subway Train Status Board, Graphing Mean Daily Plaza Traffic. Scales, Axes, and Lines, Graphing Turnstile Traffic, Interaction and Transitions, Subway Connectivity, Scheduled Wait Time Distribution	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. **Bear bibeault, Yehuda katz**: jQuery in Action. 3rd Edn, DreamTech India,2008
2. **RobertW.Sebesta**:Programming the Worldwide Web, 4th Edn, Pearson, 2012
3. **Francis Shanahan:**Mashups,WileyIndia, 2012
4. **Mike Dewar**: "Getting Started with D3": O'Reilly Media, 2012

Reference Books:

1. M.Deitel,P.J.Deitel,A.B.Goldberg:Internet &Internet & World Wide Web How to program,3rd Edition, Pearson Education/PHI, 2004

SOFTWARE TESTING AND PRACTICES

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	16MCA43	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Acquire knowledge of basic principles and knowledge of software testing and debuggingand test cases.

CO2: Understand the perceptions on testing like levels of testing, generalized pseudo code and with related examples

CO3: Study the various types of testing.

CO4: Analyze the difference between functional testing and structural testing.

CO5: Analyze the performance of fault based testing.

.

Modules	**Teaching Hours**
Module -1	**10 Hours**
Basics of Software Testing, Basic Principles, Test case selection and Adequacy 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. Sensitivity, Redundancy, Restriction, Partition, Visibility and Feedback, Test Specification and cases, Adequacy Criteria, Comparing Criteria	
Module -2	**10 Hours**
A perspective on Testing Basic definitions, Test cases, Insights from a Venn diagram, Identifying test cases, Error and fault taxonomies, Level of testing, Examples: Generalized pseudo code, The triangle problem, the Next Date function, The commission problem, The SATM (Simple Automation Teller Machine) problem, The currency converter, Saturn windshield wiper	
Module -3	**10 Hours**
Boundary value testing, Equivalence class testing, Decision table based testing Boundary value analysis, Robustness testing, Worst-case testing, special value testing, Examples, Random testing, Equivalence classes, Equivalence test cases for triangle problem, Next Date function and commission problem, Guidelines and observations, Decision tables, Test cases for triangle problem	
Module -4	**10 Hours**
Path Testing, Data flow testing, Levels of Testing, Integration Testing DD Paths, Test coverage metrics, Basis path testing, guidelines and observations, Definition Use testing, Slice based testing, Guidelines and observations. Traditional view of testing levels, Alternative life cycle models, the SATM systems, separating integration and system testing, Guidelines and observations.	

Module -5	**10 Hours**
Fault Based Testing, Planning and Monitoring the Process, Documenting Analysis and Test 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. 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.	
Question paper pattern: • The question paper will have ten questions. • Each full question consists of 16 marks. • There will be 2full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. Adithya P.Mathur " Foundations of Software Testing – Fundamental Algorithms and Techniques", Pearson Education India, 2011 2. Mauro Pezze, Michael Young, Software testing and Analysis- Process, Principles and Techniques, Wiley India, 2012 3. Paul C Jorgensen, "Software Testing A Craftsman's Approach", Auerbach publications, 3rd edition, 2011.	
Reference Books: 1. KshirasagaraNaik, PriyadarshiTripathy: Software Testing and Quality Assurance, Wiley India 2012 2. M.G.Limaye: Software Testing-Principles, Techniques and Tools – McGraw Hill, 2009	

ADVANCED JAVA PROGRAMMING LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

IV SEMESTER

Laboratory Code	16MCA46	CIE Marks	20
Number of LAB Hours/Week	01Hr Instructions 02 Hrs Laboratory	SEE Marks	80
Total Number of LAB Hours	42	SEE Hours	3 Hrs

CREDITS – 02

Course Outcome (CO): At the end of this course, the students will be able to

CO 1: Designing HTML pages to demonstrate Java Servlets, JSP, Bean and EJB programs.

CO 2: Implementing Dynamic HTML using Servlet and demonstration of service methods, auto web page refresh, Session tracking using cookie and Http Session in Servlet.

CO 3: Learn the fundamental of connecting to the database.

CO 4: Demonstrate JSP (page attributes, action tags and all basic tags) and types of EJB application.

Program Statements

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.Write a JAVA Servlet program to track HttpSession by accepting user name and password using HTML and display the profile page on successful login.

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

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

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

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

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

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 1: InthepracticalExaminationstudent has to executeoneprogramfroma lot of all the 13 questions.

Note 2: Change of program is not permitted in the Practical Examination.

ADVANCED WEB PROGRAMMING LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

IV SEMESTER

Laboratory Code	16MCA47	CIE Marks	20
Number of LAB Hours/Week	01Hr Instructions 02 Hrs Laboratory	SEE Marks	80
		SEE Hours	3 Hrs

CREDITS – 02

NOTE:

1. In the examination, student should executeone question from part A.
2. Web application project group size is limited to two students only.
3. The project under part B has to be evaluated.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand, analyze and apply the role of server side scripting languages.

CO2: Build web application using PHP, Ruby, jQuery, XML and store values in MYSQL.

CO3: Build web applications consisting of graphs using D3.JS.

CO4: Analyze a web project and identify its elements and attributes In comparison to traditional projects.

PART – A
Server Side Web Programming

1.	Write jQueryprogram to solve the following: a) Limit character input in the textarea including count. b) Based on check box, disable/enable the form submit button.
2.	a) Write a PHP program to store current date-time in a COOKIE and display the 'Last visited on' date-time on the web page upon reopening of the same page. b) Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.
3.	Write a PHP 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.
4.	Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.
5.	Write a PHP program to read student data from an XML file and store into the MySQL database. Retrieve and display.
6.	Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Provide buttons to update and delete data for the same.
7.	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.
8.	Create a XHTML form to collect Viewers Opinion based on rating (1 to 5). Finally Display the survey report in terms of a bar-graph using D3-JS

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 (Part A : 40 Marks and Part B : 40 Marks):

1. In the examination studenthas to execute one question from part A which is evaluated for 40 Marks.
2. A team of maximum two students must develop the web application 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 30 Marks.

SOFTWARE TESTING LABORATORY

[As per Choice Based Credit System (CBCS) scheme] SEMESTER – IV			
Laboratory Code	16MCA48	CIE Marks	20
Number of LAB Hours/Week	01Hr Instructions 02 Hrs Laboratory	SEE Marks	80
		SEE Hours	3 Hrs
CREDITS – 02			

Laboratory Experiments

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Analyze the performance of fault based testing, planning and Monitoring the process, Documentation testing

CO2: This course provides to experience on software testing projects using software testing tools.

CO3: Understand the process to be followed in software development life cycle.

CO4: Practical solutions to the problems

CO5: Define, formulate and analyze a problem.

1. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Derive test cases for your program based on decision-table approach, the test cases and discuss the results.
2. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on boundary-value analysis, execute the test cases and discuss the results.
3. Design and develop a program in a language of your choice to solve the triangle problem defined as follows: Accept three integers which are supposed to be the three sides of a triangle and determine if the three values represent an equilateral triangle, isosceles triangle, scalene triangle, or they do not form a triangle at all. Assume that the upper limit for the size of any side is 10. Derive test cases for your program based on equivalence class partitioning, execute the test cases and discuss the results.
4. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of dataflow testing, derive different test cases, execute these test cases and discuss the test results.
5. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of boundary value testing, derive different test cases, execute these test cases and discuss the test results.
6. Design, develop, code and run the program in any suitable language to solve the commission problem. Analyze it from the perspective of equivalence class testing, derive different test cases, execute these test cases and discuss the test results.

	Note 1: InthepracticalExaminationstudent has to executeoneprogramfroma lot of all the 6 questions. **Note 2: Change of program is not permitted in the Practical Examination.**

ADVANCED COMPUTER NETWORKS
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – IV

Subject Code	16MCA441	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Understand the terminology and concepts of TCP-IP reference model and IPV6 message format and its services.
CO2: Acquire theconceptsofprotocols,networkinterfaces,anddesign/performanceissuesin LocalAreaNetworksandwideareanetworks.
CO3: Analyze the difference between wireless networks and satellite network.
CO4: Evaluate the performance of TCP/IPoverasymmetric networks

Modules	**Teaching Hours**
Module -1	**08 Hours**
Introduction to Computer Networks History of TCP/IP, TCP Applications and Services, Performance Study of TCP/IP, earning of TCP Performance? 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, Reasons for Network Measurement, Measurement Tasks, Classification of Measurement Tools, Popular Measurement Tools and Their Applications, Tcpdump, Tcpstat, Ttcp&Netperf. Distributed Benchmark System.	
Module -2	**08 Hours**
TCP/IP Network Simulation and TCP Modeling 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. Motivation for Mathematical. Modeling of TCP, Essentials of TCP Modeling, Window Dynamics Packet-Loss Process, Gallery of TCP Models, Periodic Model, Detailed Packet Loss Model, Stochastic Model with General Loss Process, Control System Model and Network System Model.	
Module - 3	**08 Hours**
TCP/IP Performance over Wireless Networks & TCP/IP Performance over Mobile Networks & Optical Networks Layer Wireless Networks: Generic Characteristics, Wireless Local Area Networks and Cellular Communications Networks. TCP Performance Issues over Wireless Links, Inappropriate Reduction of Congestion Window, Throughput Loss in WLANs and Throughput Loss in Cellular Communication Systems. Improving TCP Performance	

over Wireless Links: Splitting TCP Connections, Snooping TCP at Base Stations, Notifying the Causes of Packet Loss, Adding Selective Acknowledgments to TCP and Comparison of Enhancement Schemes. Wireless System Evolution and TCP/IP: Trends in Cellular Communication Systems, Trends in Wireless LAN Systems, TCP/IP over Heterogeneous Wireless systems. 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. Evolution of Optical Networks, IP over DWDM, Multiprotocol Label Switching, Multiprotocol Lambda Switching, Optical Burst Switching, Optical Packet Switching: Optical Packet Format, Congestion Resolution in Optical Packet Switches, Performance of TCP/IP over Optical Networks, Optical Packet Network End-to-End Performance, Mapping of TCP in Optical Packets, Optical Packet Design in the TCP/IP Environment.	
Module - 4	**08 Hours**
TCP/IP Performance over Satellite Networks & TCP/IP Performance over Asymmetric Networks A Brief History of Data Satellites, Motivations for Using Satellites, Types of Satellites Satellite Internet Architectures, Satellite Characteristics Affecting TCP: Long Feedback Loop, Link Impairment, Bandwidth-Delay Product, Bandwidth Asymmetry, Variable Delays, LEO Handoff Spectral Congestion, Security. TCP Enhancements for Satellite Networks: Path MTU Discovery, TCP for Transactions, Window Scaling, Large Initial Window, Byte Counting, Delayed ACKs after Slow Start, Explicit Congestion Notification, Multiple Connections, Pacing TCP Segments, TCP/IP Header Compression, and Security Issues Conclusions for TCP Enhancements.Advanced Enhancements and New Versions of TCP: Quick-Start TCP, High Speed TCP ,TCP Peach, Explicit Transport Error Notification TCP Westwood and XCP .New Transport Protocols for Satellite Links: Satellite Transport Protocol, Space Communications Protocol Specifications-Transport Protocol. Types of Network Asymmetry: Bandwidth Asymmetry, Media-Access Asymmetry, Loss Rate. Asymmetry Impact of Asymmetry on TCP performance: Bandwidth Asymmetry, Media Access Asymmetry. Improving TCP Performance over Asymmetric Networks: Uplink Bandwidth Management Handling Infrequent ACK. Experimental Evaluation Of Performance Improvement Techniques Experiments with Bandwidth Asymmetry, Experiments with Media Access Asymmetry	
Module – 5	**8 Hours**
TCP/IP Performance over Asymmetric Networks & New TCP Standards and Flavors Types of Network Asymmetry: Bandwidth Asymmetry, Media-Access Asymmetry, Loss Rate. Asymmetry Impact of Asymmetry on TCP Performance: Bandwidth Asymmetry, Media Access Asymmetry. Improving TCP Performance over Asymmetric Networks: KS: Uplink Bandwidth Management Handling Infrequent ACK. Experimental Evaluation of Performance Improvement Techniques Experiments with Bandwidth Asymmetry, Experiments with Media Access Asymmetry. Duplicate Acknowledgments and Fast Retransmit, Fast Recovery and TCP Reno, TCP New Reno, TCP with Selective Acknowledgments, Forward Acknowledgments, TCP Vegas ,Overview of Other Features and Options and Performance Comparison of TCP Flavors	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. High Performance TCP/IP: Networking Concepts, Issues, and Solutions, Mahbub Hassan and Raj Jain, IST Edition, 2009 PHI Learning 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:

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

DATA WAREHOUSING AND DATA MINING

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	16MCA442	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Learn the concept of Data warehousing and OLAP.

CO2: Understand storage and retrieval technique of data from DATA CUBE.

CO3: Analyze different types of data and different preprocessing techniques.

CO4: Evaluate various Association algorithms and its applications.

CO5: Apply different Classification technique.

CO6: Evaluate different types of classifiers.

CO7: Analyze different clustering techniques and their applications

Modules	**Teaching Hours**
Module -1	**08 Hours**
Data warehousing and OLAP Data Warehouse basic concepts, Data Warehouse Modeling, Data Cube and OLAP : Characteristics of OLAP systems, Multidimensional view and Data cube, Data Cube Implementations, Data Cube operations, Implementation of OLAP and overview on OLAP Software.	
Module -2	**8 Hours**
Data Mining and its Applications Introduction, What is Data Mining, Motivating Challenges, Data Mining Tasks, Which technologies are used for data mining, Kinds of pattern that can be mined,Data Mining Applications, Data Preprocessing, Datacleaning, data integration, data reduction and data transformation.	
Module-3	**8 Hours**
Association Analysis: Basic Concepts and Algorithms 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	
Module-4	**8 Hours**
Classification : Methods, Improving accuracy of classification Basics, General approach to solve classification problem, Decision Trees, Rule Based Classifiers, Nearest Neighbor Classifiers. Bayesian Classifiers, Estimating Predictive accuracy of classification methods, Improving accuracy of classification methods, Evaluation criteria for classification methods, Multiclass Problem.	
Module-5	**8 Hours**
Clustering Techniques 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	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Jiawei Han and MichelineKamber: Data Mining - Concepts and Techniques, 2nd Edition, Morgan Kaufmann Publisher, 2006.
2. Pang-Ning Tan, Michael Steinbach, Vipin Kumar: Introduction to Data Mining, Addison- Wesley, 2005.

Reference Books:

1. Arun K Pujari: Data Mining Techniques University Press, 2nd Edition, 2009.
2. G. K. Gupta: Introduction to Data Mining with Case Studies, 3rd Edition, PHI, New Delhi, 2009.
3. Alex Berson and Stephen J.Smith: Data Warehousing, Data Mining, and OLAP Computing McGrawHill Publisher, 1997.

SOFTWARE ARCHITECTURE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	16MCA443	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Acquire knowledge of

- working principles, characteristics and basic applications of Architectural patterns.
- project life cycle context.
- how the architecture is influenced.
- the quality attributes of architecture.

CO2: Modeling quality attributes through

- check lists.
- experiments.
- back-of-the envelope analysis.

CO3: Understand the techniques of requirements gathering through interviewing stake holders, etc.

CO4: Understand different types of design patterns.

Modules	Teaching Hours
Module -1	**8 Hours**
Introduction and Context of Software Architecture What software architecture is and what it is not; Architectural Structures and views; Architectural patterns; What makes a "good" architecture? Why is software important? Technical Context; Project life-cycle context; Business context; Professional context; Stake holders; How is Architecture influenced? What Do Architecture influence?	
Module -2	**8 hours**
Understanding Quality Attributes Architecture & Requirements; Functionality; quality attribute considerations; Specifying and achieving Quality attribute requirements; Guiding quality design decisions; Availability; Interoperability; Modifiability; Performance; Security; Testability; Usability.	
Module - 3	**8 Hours**
Quality Attribute modeling and Analysis, Architecture and requirements Gathering 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. Architecture and requirements 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.	

Module - 4	**8 Hours**
Designing an Architecture, Documenting Software Architecture and Architecture, Implementation & Testing Design strategy; the attribute driven design methods; the steps of ADD, Uses and Audiences for architecture documentation; Notations, View and Behavior; Documentation and quality attributes, Architecture and implementation; Architecture and testing.	
Module -5	**8 Hours**
Architectural Patterns Introduction to patterns; From Mud to structure; Layers; Pipes and filters; Blackboard; Distributed systems; Brocker; Interactive systems; Model-view-control; Presentation- abstraction- control; Adaptable systems; Microkernel	
Question paper pattern: • The question paper will have ten questions. • Each full question consists of 16 marks. • There will be 2full questions (with a maximum of four sub questions) from each module. • Each full question will have sub questions covering all the topics under a module. • The students will have to answer 5 full questions, selecting one full question from each module.	
Text Books: 1. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, 3rd 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, RegineMeunier, 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: 1. Richard N.Taylor, NenadMedvidovic and Eric M. Dashofy: Software Architecture: Foundations, Theory, and Practice, Wiley- India 2012 2. Mary Shaw and DavidGarlan : Software Architecture - Perspectives on an Emerging Discipline, Prentice Hall of India, 2007.	

CRYPTOGRAPHY AND NETWORK SECURITY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	16MCA444	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcome (CO): At the end of this course, the students will be able to

CO1: Identify common network security vulnerabilities/attacks;

CO2: Understand the foundations of Cryptography and network security.

CO2: Understand encryption and decryption of messages using block ciphers

CO3: Demonstrate detailed knowledge of the role of encryption to protect data.

CO4: Analyze Network Security Practice And System Security.

Modules	**Teaching Hours**
Module -1	**08 Hours**
Introduction to Cryptography **Introduction:** OSI Security Architecture, Security Attacks, Security Services, Security Mechanism, model for Network Security. **Classical Encryption Technique:** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques.	
Module -2	**8 Hours**
Data Encryption and advanced encryption techniques **Block Ciphers, Data Encryption Standard and Advanced Encryption Standard** Block Cipher Principles, The Data Encryption Standard, Block Cipher Design Principles and Modes of operation, Evaluation Criteria for AES, AES Cipher-Encryption and Decryption, Data Structure, Encryption Round. **Public Key Cryptography and Key Management** Principles of Public Key Cryptosystem, RSA algorithm, Key management, Diffie Hellman Key exchange.	
Module - 3	**8 Hours**
CRYPTOGRAPHY techniques **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	
Module - 4	**8 Hours**
E-MAIL AND IP SECURITY **Electronic Mail Security:** Pretty Good Privacy (PGP), S/MIME **IP Security:** IP Security Overview;IP Security Architecture; Authentication Header; Encapsulating SecurityPayload; Combining Security Associations; Key Management.	
Module – 5	**8 Hours**
WEB AND SYSTEM SECURITY **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.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. William Stallings, “Cryptography and Network Security – Principles and Practices”, 4th Edition, Pearson Education, 2009. (Chapters: 1, 2.1-2.3, 3.1,3.2,3.5, 5.1,5.2, 6.2, 9.1,9.2, 10.1,10.2, 11.1-11.4, 13.1, 13.3, 14.1, 4.2, 15.1, 15.2, 16.1-16.6, 17.1-17.3, 18.1, 18.2, 20.1; Exclude the topic not mentioned in the syllabus)

Reference Books:

1. Behrouz A. Forouzan and DebdeepMukhopadhyay: “Cryptography and Network Security”, 2nd Edition, Tata McGraw-Hill, 2010.

2. AtulKahate, “Cryptography and Network Security” 2nd Edition TMH.

WIRELESS COMMUNICATION&MOBILE TECHNOLOGIES
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – IV

Subject Code	16MCA451	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course outcomes:*At the end of this course, the students will be able to*

CO1: Understand the classification of devices, limitations of devices, interfaces, protocol and system Architecture

CO2: Understand the importance of Wireless Medium Access control and CDMA based communication and its Applications.

CO3: Understand the concepts of Network layer, Transport layer.

CO4: Analyze the working of Data Dissemination and Broadcasting Systems.

CO5: Understand and apply the Data Synchronization Server and Management Application languages (XML, Java, J2ME and JavaCard, Mobile Operating Systems).

Modules	**Teaching Hours**
Module -1	**8 Hours**
Mobile Devices and Systems, Architectures Mobile phones, Digital Music Players, Handheld Pocket Computers, Handheld Devices, Operating Systems, Smart Systems, Limitations of Mobile Devices, Automotive Systems GSM – Services and System Architectures, Radio Interfaces, Protocols, Localization, Calling, Handover.	
Module -2	**8 Hours**
Wireless Medium Access Control and CDMA – based Communication Medium Access Control, Introduction to CDMA – based Systems. OFDM	
Module - 3	**8 Hours**
Mobile IP Network Layer Mobile Transport Layer and Databases Packet Delivery and Handover Management, Registration, Tunneling and Encapsulation, Route Optimization, Dynamic Host Configuration Protocol Indirect TCP, Snooping TCP, Mobile TCP, Other Methods of TCP – layer Transmission for Mobile Networks. **Databases** Database Hoarding Techniques, Data Caching, Client – Server Computing and Adaptation, Transactional Models, Query Processing, Data Recovery Process	
Module -4	**8 Hours**
Data Dissemination and Broadcasting Systems Communication Asymmetry, Classification of Data – Delivery Mechanisms, Data Dissemination Broadcast Models, Selective Tuning and Indexing Techniques, Digital Audio Broadcasting. Digital video Broadcasting.	
Module -5	**8 Hours**
Data Synchronization in Mobile Computing Systems Synchronization, Synchronization Protocols, SyncML – Synchronization Language for Mobile Computing. Mobile Devices, Server and Management, Wireless LAN, Mobile Internet Connectivity and Personal Area NetworkMobile agent, Application Server, Gateways, Portals, Service Discovery, Device Management, Mobile File Systems. Wireless LAN	

(WiFi) Architecture and Protocol Layers, WAP 1.1 and WAP 2.0 Architectures, Bluetooth – enabled Devices Network, Zigbee.
Mobile Application languages – XML, Java, J2ME and JavaCard, Mobile Operating SystemsIntroduction, XML, JAVA, Java 2 Micro Edition (J2ME), JavaCard Operating System, PalmOS, Windows CE, Symbian OS.

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Raj Kamal: Mobile Computing, Oxford University Press, 2007.

Reference Books:

1. Ashok Talukdar, Roopa R Yavagal: Mobile Computing – Technology, Applications and Service Creation, Tata McGraw Hill, 2005.
2. 2 Reza B'Far: Mobile Computing Principles – Designing and Developing Mobile Applications with UML and XML, 5th Edition, Cambridge University press, 2006.
3. UweHansmann, LothatMerk, Martin S Nicklous and Thomas Stober: Principles of Mobile Computing, 2nd Edition, Springer International Edition, 2003.
4. Schiller: Mobile Communication, Pearson Publication, 2004.

BIG DATA ANALYTICS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – IV

Subject Code	16MCA452	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course outcomes (CO):***At the end of this course, the students will be able to:***

CO1: Understand the Map Reduce technique for solving Big Data problems

CO2: Understand algorithms for Big Data by deciding on the apt Features set

CO3: Apply algorithms for handling peta bytes of datasets

CO4: Analyze main memory consumption for Big Data analytics

CO5: Understand and analyze the usage of map reduce techniques for solving big data problems

Modules	**Teaching Hours**
Module -1	**8 Hours**
Big Data and Analytics Example Applications, Basic Nomenclature, Analysis Process Model, Analytical Model Requirements , types of Data Sources, Sampling, Types of data elements, data explorations, exploratory statistical analysis, missing values, outlier detection and Treatment, standardizing data labels, categorization	
Module -2	**8 Hours**
Big Data Technology Hadoop's Parallel World – Data discovery – Open source technology for Big Data Analytics – cloud and Big Data –Predictive Analytics – Mobile Business Intelligence and Big Data – Crowd Sourcing Analytics – Inter- and Trans-Firewall Analytics	
Module -3	**8 Hours**
Meet Hadoop Data, Data Storage and Analysis ,Comparison with Other Systems,RDBMS,Grid Computing Volunteer Computing, A Brief History of Hadoop,ApacheHadoop and the Hadoop Ecosystem Hadoop Releases Response	
Module -4	**8 Hours**
The Hadoop Distributed File system The Design of HDFS, HDFS Concepts, Blocks, Namenodes and Datanodes, HDFS Federation, HDFS High-Availability, The Command-Line Interface, Basic Filesystem Operations, HadoopFilesystems Interfaces ,The Java Interface, Reading Data from a Hadoop URL, Reading Data Using the FileSystem API, Writing Data, Directories, Querying the Filesystem, Deleting Data, Data Flow Anatomy of a File Read ,Anatomy of a File Write, Coherency Model, Parallel Copying with distcp Keeping an HDFS Cluster Balanced, Hadoop Archives	
Module -5	**8 Hours**
Map Reduce A Weather Dataset ,Data Format, Analyzing the Data with Unix Tools, Analyzing the Data with Hadoop, Map and Reduce, Java MapReduce, Scaling Out, Data Flow, Combiner functions, Running a Distributed MapReduce Job, Hadoop Streaming, Hadoop Pipes, Compiling and Running, Developing a MapReduce Application, The Configuration API, Combining Resources, Variable Expansion, Configuring the Development Environment, Managing Configuration, GenericOptionsParser, Tool and	

ToolRunner, Writing a Unit Test, Mapper, Reducer, Running Locally on Test Data, Running a Job in a Local Job Runner, Testing the Driver, Running on a Cluster, Packaging, Launching a Job, The MapReduce Web UI, Retrieving the Results, Debugging a Job, Hadoop Logs, Remote Debugging.

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Bart Baesens, " Analytics in a Big Data World : The Essential Guide to Data Science and its Applications" Wiley
2. Michael Minelli, Michehe Chambers, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", 1st Edition, Michael Minelli, Michele Chambers, AmbigaDhiraj, Wiley CIO Series, 2013.
3. Tom White, "Hadoop: The Definitive Guide", 3rd Edition, O'reilly, 2012

Reference Books:

1. Boris lublinsky, Kevin t. Smith, Alexey Yakubovich, "Professional Hadoop Solutions", Wiley, ISBN: 9788126551071, 2015.
2. Chris Eaton, Dirk deroos et al. , "Understanding Big data ", McGraw Hill, 2012.
3. VigneshPrajapati, "Big Data Analytics with R and Haoop", Packet Publishing 2013.
4. Tom Plunkett, Brian Macdonald et al, "Oracle Big Data Handbook", Oracle Press, 2014.

SOFTWARE QUALITY MANAGEMENT
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – IV

Subject Code	16MCA453	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course outcomes:***At the end of this course, the students will be able to:***

- **CO1:**Understand the Software Quality through Hierarchical models.
- **CO2:**Design the quality software by SQA plan Reviews and Audits.
- **CO3:**Apply quality control through CASE tools.
- **CO4:**Understand different quality standards.

Modules	**Teaching Hours**
Module -1	**8 Hours**
INTRODUCTION TO SOFTWARE QUALITY Software Quality - Hierarchical models of Boehm and McCall – Quality measurement – Metrics measurement – Metrics measurement and analysis – Gilb’s approach – GQM Model	
Module -2	**8 Hours**
SOFTWARE QUALITY ASSURANCE Quality tasks – SQA plan – Teams – Characteristics Implementation – Documentation– Reviews and Audits.	
Module - 3	**8 Hours**
QUALITY CONTROL AND RELIABILITY Tools for Quality – Ishikawa’s basic tools – CASE tools Defect prevention and removal – Reliabilitymodels, Rayleigh model – Reliability growth models for quality assessment.	
Module -4	**8 Hours**
QUALITY MANAGEMENT SYSTEM Elements of QMS – Rayleigh model framework Reliability Growth models for QMS – Complexity metrics and models Customer satisfaction analysis.	
Module -5	**8 Hours**
QUALITY STANDARDS Need for standards – ISO 9000 Series – ISO 9000 3 for software development – CMM and CMMI – Six Sigma concepts.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Allan C. Gillies, “Software Quality: Theory and Management”, Thomson Learning, 2003.
2. Stephen H. Kan, “Metrics and Models in Software Quality Engineering”, Pearson Education (Singapore) Pte Ltd., 2002.

Reference Books:

1. Norman E. Fenton and Shari Lawrence Pfleeger, “Software Metrics” Thomson, 2003
2. Mordechai Ben Menachem and Garry S.Marliss, “*Software Quality*”, Thomson Asia Pvt Ltd, 2003
3. Mary Beth Chrissis, Mike Konrad and Sandy Shrum, “*CMMI* ”, Pearson Education (Singapore) Pvt Ltd, 2003

Principles of User Interface Design

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER –IV

Subject Code	16MCA454	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcome (CO): At the end of this course, the students will be able to

CO1:usethe new technologies that provide interactive devices and interfaces.

CO2:apply the process and evaluate UID.

CO3:understand Direct Manipulation and Virtual Environment

CO4:discuss the command, natural languages and issues in design for maintaining QoS.

CO5:persuade user documentations and information search.

Modules	**Teaching Hours**
Module -1	**8 Hours**
Introduction 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.	
Module -2	**8 Hours**
Development Processes Managing Design Processes: Introduction, Organizational Design to support Usability, The Four Pillars of Design, Development methodologies: Ethnographic Observation, Participatory Design, Scenario Development, Social Impact statement for Early Design Review, Legal Issues. **Evaluating Interface Design** Introduction, Expert Reviews, Usability Testing and Laboratories, Survey Instruments, Acceptance tests, Evaluation during Active Use, Controlled Psychologically Oriented Experiments	
Module -3	**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	
Module -4	**8 Hours**
Command and Natural Languages 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

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

Module -5	**8 Hours**

User Documentation and Online Help :

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

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

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Ben Shneiderman, Plaisant, Cohen, Jacobs: Designing the User Interface, 5th Edition, Pearson ,Education, 2010.

Reference Books:

1 Alan Dix, Janet Finalay, Gregory D AbiwdmRusselBealel: Human-Computer Interaction, III Edition, Pearson , Education, 2008.

2 Eberts: User Interface Design, Prentice Hall, 1994

3 Wilber O Galitz: The Essential Guide to User Interface Design- An Introduction to GUI Design, Principles and Techniques, Wiley-Dreamtech India Pvt Ltd, 2011

OBJECT-ORIENTED MODELING AND DESIGN PATTERNS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	16MCA51	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Acquire knowledge of

- Basic UML Concepts and terminologies
- Life Cycle of Object oriented Development
- Modeling Concepts

CO2: Identify the basic principles of Software modeling and apply them in real world applications

CO3: Produce conceptual models for solving operational problems in software and IT environment using UML

CO4: Analyze the development of Object Oriented Software models in terms of

- Static behaviour
- Dynamic behaviour

CO5: Evaluate and implement various Design patterns

Modules	**Teaching Hours**
Module -1	**10 Hours**
Modeling Concepts & Class Modeling 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. 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;	
Module -2	**10 Hours**
State Modeling and Interaction 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; Interaction Modeling: Use case models; Sequence models; Activity models. Use case relationships; Procedural sequence models; Special constructs for activity models	
Module -3	**10 Hours**
System Conception and Analysis System Conception: Devising a system concept; elaborating a concept; preparing a problem statement. 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.	
Module -4	**10 Hours**
System Design and Class Design Overview of system design; Estimating performance; Making a reuse plan; Breaking a system in to sub-systems; Identifying concurrency; Allocation of sub-systems; Management of data storage; Handling global resources; Choosing a software control strategy; Handling boundary conditions Class Design: Overview of class design; Bridging the gap; Realizing use cases; Designing algorithms; Recording downwards, Refactoring; Design optimization; Reification of behavior.	
Module -5	**10 Hours**
Patterns and Design Patterns What is a pattern and what makes a pattern? Pattern categories; Relationships between patterns; Pattern description. Introduction, Model View Controller, Structural decomposition: Whole-Part, Access Control: Proxy; Management Patterns: Command processor; Communication Patterns: Forwarder-Receiver; Client-Dispatcher-Server; Publisher-Subscriber	
Question paper pattern: The question paper will have ten questions. 1. Each full question consists of 16 marks. 2. There will be 2 full questions (with a maximum of four sub questions) from each module. 3. Each full question will have sub questions covering all the topics under a module. 4. The students will have to answer 5 full questions, selecting one full question from each module.	

Text Books:

1. Michael Blaha, James Rumbaugh, “Object-Oriented Modeling and Design with UML”, 2nd Edition, Pearson Education / PHI, 2005. (Chapters 1 to 9, 11 to 14.10,15.1 to 15.8)
2. Frank Buschmann, RegineMeunier, Hans Rohnert, Peter Sommerlad, Michael Stal, “Pattern-Oriented Software Architecture, A System of Patterns”, Volume 1, John Wiley and Sons, 2006. (Chapters 1, 2.4, 3)

Reference Books:

1. Grady Booch et al, “Object-Oriented Analysis and Design with Applications”, 3rd Edition, Pearson, 2007.
2. Mark Priestley, “Practical Object-Oriented Design with UML”, 2nd Edition, Tata McGraw-Hill, 2003.
3. K. Barclay, J. Savage, “Object-Oriented Design with UML and JAVA”, Elsevier, 2008.
4. Booch, G., Rumbaugh, J., and Jacobson, I., “The Unified Modeling Language User Guide”, 2nd Edition, Pearson, 2005.
5. E. Gamma, R. Helm, R. Johnson, J. Vlissides, “Design Patterns-Elements of Reusable Object- Oriented Software”, Addison-Wesley, 1995.
6. Michael R Blaha, James R Rumbaugh, “Object Oriented Modeling and Design with UML”, 2nd Edition, Prentice Hall, 2004

PROGRAMMING USING C#&.NET

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	16MCA52	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Understand C# and client-server concepts using .Net Frame Work Components.
CO2: Apply delegates, event and exception handling to incorporate with ASP, Win Form, ADO.NET.
CO3: Analyze the use of .Net Components depending on the problem statement.
CO4: Implement & develop a web based and Console based application with Database connectivity

Modules	**Teaching Hours**
Module -1	**10 Hours**
Getting started with .NET Framework 4.0 and C# Understanding Previous Technologies, Benefits of .NET Framework, Architecture of .NET Framework 4.0,.NET Execution Engine, Components of .NET Framework 4.0: CLR, CTS, Metadata and Assemblies, .NET Framework Class Library, Windows Forms, ASP .NET and ASP .NET AJAX, ADO .NET, Windows workflow Foundation, Windows Presentation Foundation, Windows Communication Foundation, Widows Card Space and LINQ. **Introducing C#** Creating a Simple C# Console Application, Identifiers and Keywords. System Data Types, Variables and Constants: Value Types, Reference Types, Understanding Type Conversions, Boxing and UnBoxing. Namespaces, The System namespace, .NET Array Types	
Module -2	**10 Hours**
Classes, Objects and Object Oriented Programming Classes and Objects: Creating a Class, Creating an Object, Using this Keyword, Creating an Array of Objects, Using the Nested Classes, Defining Partial Classes and Method, Returning a Value from a Method and Describing Access Modifiers. Static Classes and StaticMembers, Properties: Read-only Property, Static Property, Indexers, Structs: Syntax of a struct and Access Modifiers for structs, System.Object Class	

Encapsulation: Encapsulation using accessors and mutators, Encapsulation using Properties. Inheritance: Inheritance and Constructors, Sealed Classes and Sealed Methods, Extension methods. **Polymorphism:** Compile time Polymorphism/ Overloading, Runtime Polymorphism/ Overriding. Abstraction: Abstract classes, Abstract methods. Interfaces: Syntax of Interfaces, Implementation of Interfaces and Inheritance.	
Module -3	**10 Hours**
Delegates, Events, Exception Handling and ADO.NET **Delegates:**Creating and using Delegates, Muticasting with Delegates. Events: Event Sources, Event Handlers, Events and Delegates, Multiple Event Handlers. **Exception Handling**: The try/catch/throw/finally statement, Custom Exception.System.Exception, Handling Multiple Exception **Data Access with ADO.NET :**Understanding ADO.NET: Describing the Architecture of ADO.NET, ADO.NET,ADO.NET Entity Framework. Creating Connection Strings: Syntax for Connection Strings.Creating a Connection to a Database: SQL Server Database, OLEDB Database, ODBC Data Source. Creating a Command Object. Working with DataAdapters: Creating DataSet from DataAdapter.	
Module -4	**10 Hours**
Graphical User Interface with Windows Forms and WPF **Windows Forms :** Introduction, Windows Forms, Event Handling: A Simple Event-Driven GUI, Control Properties and Layout, Labels, TextBoxes and Buttons, GroupBoxes and Panels, CheckBoxes and RadioButtons, ToolTips, Mouse-Event Handling, Keyboard-Event Handling. Menus, Month Calendar Control, LinkLabel Control, ListBox Control, ComboBox Control, TreeView Control, ListView Control, TabControl and Multiple Document Interface (MDI) Windows. **WPF:** New WPF Controls**,** WPF Architecture: Presentation Framework, Presentation Core, WindowsBase, MIL or Milcore, Working with WPF Windows: Using XAML in WPF 4.0 Applications : Contents of XAML and WPF Applications: XAML Elements Namespace and XAML, XAML Property Syntax, Markup Extensions.	
Module -5	**10 Hours**
Web App Development and Data Access using ADO.NET Introduction, Web Basics, Multitier Application Architecture, Your First Web Application: Building Web-Time Application, Examining Web-Time.aspx's Code-Behind File, Understanding Master pages, Standard Web Controls: Designing a Form, Validation Controls, GridView Control, DropDownList, Session Tracking, **ASP.NET AJAX :** Exploring AJAX,Need for AJAX, AJAX and other Technologies, AJAX Server Controls, ScriptManager control, Update Panel, UpdateProgress Control, Creating Simple Application using AJAX Server Controls	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

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).
2. Paul Deitel and Harvey Deitel: C# 2010 for Programmers, 4th Edition, Pearson Education. (Chapters: 14,15,19 and 27.3)

Reference Books:

1. Andrew Trolsen: Pro C# 5.0 and the .NET 4.5 Framework, 6th Edition, Wiley-Appress.
2. Bart De Smet: C# 4.0 Unleashed, Pearson Education- SAMS Series.
3. Herbert Schildt: Complete Reference C# 4.0, Tata McGraw Hill, 2010.

MOBILE APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – V

Subject Code	16MCA53	CIE Marks	20
Number of Lecture Hours/Week	04	SEE Marks	80
Total Number of Lecture Hours	50	SEE Hours	03

CREDITS – 04

Course Outcomes(CO):*At the end of this course, the students will be able to*

CO1: Illustrate effective user interfaces that leverage evolving mobile device capabilities
CO2: Develop applications using software development kits (SDKs), frameworks and toolkits
CO3: Establish various methods to integrate database and server-side technologies
CO4: Design and develop open source software based mobile applications
CO5: Build and deploy competent mobile development solutions

Modules	**Teaching Hours**
Module -1	**10 Hours**
Introduction Preliminary Considerations – Cost of Development – Importance of Mobile Strategies in the Business World – Effective use of Screen Real Estate – **Understanding Mobile Applications** Understanding Mobile Applications Users – Understanding Mobile Information Design – Understanding Mobile Platforms – Using the Tools of Mobile Interface Design.	
Module -2	**10 Hours**
Getting Started with Android Programming What is Android – Obtaining the required tools– Anatomy of an Android Application – Components of Android Applications – Activities – Fragments – Utilizing the Action Bar	
Module -3	**10 Hours**
Android UI Design and Location Based Services Views and View Groups – Basic Views – Fragments – Displaying Maps – Getting Location Data – Publishing for Publishing – Deploying APK Files	
Module -4	**10 Hours**
Android Messaging and Networking SMS Messaging – Sending Email – Networking – Downloading Binary Data, Text files – Accessing Web Services – Performing Asynchronous Call – Creating your own services – Communicating between a service and an activity – Binding activities to services	
Module -5	**10 Hours**
Feedback and Oscillator Circuits iOS – Obtaining the tools and SDK – Components of XCODE – Architecture of iOS – Building Derby App in iOS – Other useful iOS things – Windows Phone: Getting the tools you need – Windows Phone 7 Project – Building Derby App in Windows Phone 7 – Distribution – Other useful Windows Phone Thing	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.

- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", 1st Edition, 2012, *ISBN*: 978-1-118-20390-3
2. Wei-Meng Lee, "Beginning Android Application Development", Wiley 2011.

Reference Books:

1. Reto Meier, "Professional Android 4 Application Development", Wrox Publications 2012

SOFTWARE DESIGN LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	16MCA56	CIE Marks	20
Number of Practical Hours/Week Number of Instructional Hours/Week	02 01	SEE Marks	80
Total Number of Practical Hours	42	SEE Hours	03

CREDITS – 02

Laboratory Experiments:

The student has to draw the necessary UML diagrams using any suitable UML drawing tool and implement a program in Java or C++ or C# to demonstrate the design pattern specified by the examiner. For analysis and design models -**, Class Diagram, Use-Case, Sequence diagrams** should be drawn.

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Understand the fundamental principles of Object-Oriented analysis, design, development and programming

CO2: Demonstrate and represent the UML model elements, to enable visual representation of the system being developed

CO3: Implement object oriented design model with the help of modern tool, Rational software Architect

CO4: Analyze and differentiate the static and dynamic behavior of the system for achieving the intended functionalities of the system

CO5: Evaluate Various design patterns for applicability, reasonableness, and relation to other design criteria

NOTE: ANY SUPPORTING TOOL MAY BE USED.

1. **Publisher-Subscriber:**Define a one-to-many dependency between objects so that when one object changes state all its dependents are notified and updated
automatically. The Observer pattern is also known as Dependents, Publisher-Subscriber. The need to maintain consistency between related objects without making classes tightly coupled.

2. **Command Processor:** The command processor design pattern separates the request for Service from its execution. A command processor component manages request as separate objects, schedules their execution and provides additional service.

3. **Forwarder-Receiver:** Distributed peers collaborate to solve a particular problem. A peer may act as a Client, requesting services, as a server, providing services, or both. The forwarder-receiver design pattern provides transparent inter process communication for software system with a peer-to-peer interaction model. It introduces forwarder and receiver to decouple peers from the underlying communication mechanism.

4. **Client-Dispatcher:**Provide a dispatcher component to act as an intermediate layer between clients and servers. The dispatcher implements a name service that allows client to refer to servers by name instead of physical locations, thus providing transparency. Each server is uniquely identified

by its name and is connected to clients by the dispatcher. Client relay on the dispatcher to locate a particular server and to establish a communication like with the server.

5. **Proxy:** A proxy object can act as the intermediary between the client and target object. The proxy object has the same interface as the target object. The proxy holds a reference to the target object and can forward requests to target as required. In effect the proxy object has the authority the act on behalf of client to interact with the target object.

6. **Polymorphism:** When related alternatives or behaviors vary by type (class), assign responsibility for the behavior using polymorphic operations to the types for which the behavior varies.This interface will provide the behavior which varies according to the class type. All classes implementing this interface will write the method accordingly.

7. **Whole-Part:** The idea of the Whole-Part pattern is to introduce a component that encapsulates smaller objects, and prevents clients from accessing these constituent parts directly. Define an interface for the aggregate that is the only means of access to the functionality of the encapsulated objects, allowing the aggregate to appear as a semantic unit.

8. **Controller Design Pattern:** Use a controller as the initial point of contact for handling a request. The controller manages the handling of the request, including invoking security services such as authentication and authorization, delegating business processing, managing the choice of an appropriate view, handling errors, and the selection of content creation strategies.

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfroma lot of all the 8 questions.

Note 2: Change of program is not permitted in the Practical Examination.

Note 3: In the Examination partial marks can be given for Class Diagram, Use-Case, Sequence diagrams

.NET LABORATORY

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	16MCA57	CIE Marks	20
Number of Practical Hours/Week Number of Instructional Hours/Week	02 01	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03

CREDITS – 02

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 50 marks and Part B has to be evaluated for 30 marks.

Course Outcomes : ***At the end of this course, the students will be able to***

CO1: Understand C# and client-server concepts using .Net Frame Work Components
CO2: Apply delegates, event and exception handling to incorporate with ASP, Win Form, ADO.NET
CO3: Analyze the use of .Net Components depending on the problem statement
CO4: Implement & develop a web based and Console based application with Database connectivity

PART-A

1.Write a Program in C# to demonstrate Command line arguments processing for the following.
a) To find the square root of a given number.
b) To find the sum & average of three numbers.

2.Write a Program in C# to demonstrate the following
a) Boxing and Unboxing b) Invalid Unboxing.

3. Write a program in C# to add Two complex numbers using Operator overloading .

4. Write a Program in C# to find the sum of each row of given jagged array of 3 inner arrays.

5. Write a Program in C# to demonstrate Array Out of Bound Exception using Try, Catch and Finally blocks.

6.Write a Program to Demonstrate Use of Virtual and override key words in C# with

a simple program.

7. Write a Program in C# to create and implement a Delegate for any two arithmetic operations

8. Write a Program in C# to demonstrate abstract class and abstract methods in C#.

9. Write a program to Set & Get the Name & Age of a person using Properties of C# to illustrate the use of different properties in C#.

10. Write a Program in C# Demonstrate arrays of interface types (for runtime polymorphism).

PART-B

I. Consider the Database db_EMS (Employee Management System)
consisting of the following tables :
tbl_Designations (IdDesignation: int, Designation: string)
tbl_EmployeeDetails(IdEmployee: int, EmployeeName: string, ContactNumber: string, IdDesignation: int, IdReportingTo: int)

Develop a suitable window application using C#.NET having following options.

1. Enter new Employee details with designation & Reporting Manager.
2. Display all the Project Leaders (In a Grid) reporting to selected Project Managers (In a Combo box).
3. Display all the Engineers (In a Grid) reporting to selected Project Leader (In a Combo box).
4. Display all the Employees (In a Grid) with their reporting Manager (No Value for PM).

NOTE: tbl_Designation is a static table containing the following Rows in it.

1	Project Manager
2	Project Leader
3	Engineer

II. Consider the Database db_LSA (Lecturer Subject Allocation) consisting of the following tables:
tbl_Subjects(IdSubject: int, SubjectCode: string, SubjectName: string)
tbl_Lecturers(IdLecturer: int, LecturerName: string, ContactNumber: string)
tbl_LecturerSubjects(IdSubject: int, SubjectCode: string, IdLecturer: int)

Develop a suitable window application using C#.NET having following options.

1. Enter new Subject Details.
2. Enter New Lecturer Details.
3. Subject Allocation with Lecturer Name in a Combo box and subjects to be allocated in Grid with check Column.
4. Display all the subjects allocated (In a Grid) to the selected Lecturer (In a Combo Box).

III. Consider the database db_VSS (Vehicle Service Station) consisting of the following tables:
tbl_VehicleTypes(IdVehicleType: int, VehicleType: string, ServiceCharge: int)
tbl_ServiceDetails(IdService: int, VehicleNumber: string, ServiceDetails: string, IdVehicleType: int)
Develop a suitable window application using C#.NET having following options.

1. Enter new Service Details for the Selected Vehicle Type (In a Combo Box).
2. Update the Existing Service Charges to Database.
3. Total Service Charges Collected for the Selected Vehicle (In a Combo box) with total amount displayed text box.

NOTE: tbl_VehicleType is a static table containing the following Rows in it.

1	Two Wheeler	500
2	Four Wheeler	1000
3	Three Wheeler	700

IV. Develop a web application using C#.NET and ASP.NET for the Postal System Management. The master p

should contain the hyper links for adding ***Area Details***, ***Postman details***, ***Letter distributions*** and ***V***
Letters.

Consider the database db_PSM (Postal System Management) consisting of the following tables:

tbl_AreaDetails(IdArea: int, AreaName: string)

tbl_PostmanDetails(IdPostman: int, PostmanName: string, ContactNumber: string, IdArea: int)

tbl_AreaLetters(IdLetter: int, LetterAddress: string, IdArea: int)

Develop the suitable content pages for the above created 4 hyper links with the following details:

1. Enter New Area Details
2. Enter New Postman Details with the Area he/she is in-charge of (display Area in a Combo box)
3. Enter all the Letters distributed to the selected Area (display Area in a Combo box)
4. Display all the Letter addresses (In a Grid) to be distributed by the selected Postman (In a Combo box)

V. Develop a web application using C#.NET and ASP.NET for the Complaint Management System. The ma page should contain the hyper links for ***Add Engineer***, ***Complaint Registration, Complaint Allocation*** ***View Complaints***.

Consider the database db_CMS (Complaint Management System) consisting of the following tables:

tbl_Departments(IdDepartment: int, DepartmentName: string)

tbl_Engineers(IdEngineer: int, EngineerName: string, ContactNumber: string, IdDepartment: int)

tbl_RegisteredComplaints(IdComplaint: int, ComplaintDescription: string)

tbl_DepartmentComplaints(IdDepartment: int, IdComplaint: int)

Develop the suitable content pages for the above created 4 hyper links with the following details:

1. Enter New Engineers belonging to the selected department (displayed in a combo box)
2. Register a new Complaint with a submit button.
3. View all registered complaints & allocate to the corresponding department (displayed in a combo bo
4. Display all the Complaints (In a Grid) to be handled by the selected Engineer (In a Combo box)

NOTE: Consider the table tbl_Departments as a static table containing some pre-entered departme which are displayed in all the remaining modules.

Note 1: InthepracticalExaminationeachstudent has to pick onequestionfromPart A and Part B each.

Note 2: Change of program is not permitted in the Practical Examination.

MINI PROJECT MOBILE APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	16MCA58	CIE Marks	20
Number of Practical Hours/Week Number of Instructional Hours/Week	02 01	SEE Marks	80
Total Number of Lecture Hours	42	SEE Hours	03

CREDITS – 03

Laboratory Programs:

The laboratory can be carried out only using any mobile application software.

Note:

1. Students are required to execute one question from Part A and give demo from Part B.
2. Part A has to be evaluated for 40 marks and Part B has to be evaluated for 40 marks along with the report.
3. The project should be carried out with a team strength of maximum two.
4. Students are expected to work for mini project apart from lab hours also with the contact of guides.

Course outcomes:*At the end of this course, the students will be able to*

- Illustrate effective user interfaces that leverage evolving mobile device capabilities
- Develop applications using software development kits (SDKs), frameworks and toolkits
- Establish various methods to integrate database and server-side technologies
- Design and develop open source software based mobile applications
- Build and deploy competent mobile development solutions

PART – A

1. Design an application that contains Phone Contacts in vertical linear manner. Selected contact appears at the top of the list with a large italicized font and a blue background.

2. Create an application that uses Layout Managers and Event Listeners.

3. Develop a standard calculator application to perform basic calculations like addition, subtraction, multiplication and division.

4. Devise an application that draws basic graphical primitives (rectangle, circle) on the screen.

5. Build an mobile application that create, save, update and delete data in a database.

6. Devise an application that implements Multi threading.

7. Develop a mobile application that uses GPS location information.

8. Create an application that writes data to the SD card.

9. Implement an application that creates an alert upon receiving a message.

10. Devise a mobile application that creates alarm clock.

PART – B

Mini-Project

Students should be able to build a complete mobile app using multiple features learnt in Part – A with user interfaces and database connectivity.

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

The report must be evaluated for 10 Marks. Demonstration and Viva for 30 Marks.

WEB 2.0 AND RICH INTERNET APPLICATIONS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	16MCA541	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcomes(CO):*At the end of this course, the students will be able to*

CO1: Understand the rich internet concepts and applications.
CO2: Analyze the working of development models in web designing.
CO3: Illustrate appropriate component lifecycle techniques using frameworks.
CO4: Evaluate and implement state based systems using data models and data binding

Modules	**Teaching Hours**
Module -1	**8 Hours**
Rich Internet Applications With Ajax: Limitations of Classic Web application model, AJAX principles, Technologies behind AJAX, Examples of usage of AJAX; Asynchronous communication and AJAX application model. **Ajax with XMLHTTP object: Part 1** 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.	
Module -2	**8 Hours**
Ajax with XMLHTTP object: Part 2 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.	
Module -3	**8 Hours**
Working with XML DOM in Ajax 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.	

Working with PHP and Ajax 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.	
Module -4	**8 Hours**
Introduction to Bootstrap. What Is Bootstrap? Bootstrap File Structure, Basic HTML Template, Global Styles, Default Grid System, Basic Grid HTML, Offsetting Columns, Nesting Columns, Fluid Grid System, Container Layouts, Responsive Design. Typography, Emphasis Classes, Lists, Code, Tables, Optional Table Classes, Table Row Classes, Forms, Buttons, Images, Icons.	
Module -5	**8 Hours**
Bootstrap Layout components. Dropdown Menus, Button Groups, Button Groups as Radio Buttons and Checkboxes, Buttons with Dropdowns, Split Button Dropdowns, Dropup Menus, Navigation Elements, Tabular Navigation, Basic Pills Navigation, Stackable Navigation, Dropdowns, Navigation Lists, Tabbable Navigation, Navbar, Forms, Navbar Menu Variations, Breadcrumbs, Pagination, Pager, Labels, Badges, Typographic Elements. **Bootstrap Javascript Plugins.** Programmatic API, Transitions, Modal, Dropdown, Dropdown Usage via JavaScript, Scrollspy, Toggleable Tabs, Tooltips, Popover, Alerts, Buttons, Collapse, Carousel, Typeahead, Affix.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Professional AJAX – Nicholas C Zakas et al, Wrox publications, 2008.
2. Steven Holzner: Ajax: A Beginner's Guide, Tata McGraw Hill, 2014.
3. Jake Spurlock: "Bootstrap: Responsive Web Development", O'Reilly Media, 2014.

Reference Books:

1. Thomas A. Powel: Ajax The Complete reference, McGraw Hill,2008.
2. AravindShenoy, Ulrich Sossou: Learning Bootstrap, Packt, Dec 2014.
3. Dana Moore, Raymond Budd, Edward Benson: Professional Rich Internet Applications: AJAX and Beyond, Wiley 2012.

CLOUD COMPUTING

[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – V

Course Code	16MCA542	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course outcomes:*At the end of this course, the students will be able to*

CO1: Understand the cloud computing delivery model and the enabling technologies.
CO2: Understand the cloud computing platforms, key technology drivers and cloud programming/software environments
CO3: Identify the need for cloud computing model and compare various key enabling technologies.
CO4: Analyze and choose an appropriate programming environment for building cloud applications.

Modules	Teaching Hours
Module -1	**08 Hours**
Distributed System Models and Enabling Technologies Scalable Computing Service over the Internet, System Models for Distributed and Cloud Computing	
Module -2	**08 Hours**
Software Environments for Distributed Systems and Clouds, Performance, Security and Energy Efficiency	
Module -3	**08 Hours**
Virtual Machines and Virtualization of Clusters and Data Centers Implementation Levels of Virtualization, Virtualization Structures /Tools and Mechanisms, Virtual Cluster and Resource Management, Virtualization for Data-Center Automation.	
Module -4	**08 Hours**
Cloud Platform Architecture over Virtualized Data Centers Cloud Computing and Service Models, Data-Center Design and Interconnection Networks, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms: GAE, AWS, and Azure, Cloud Security and Trust Management	
Module -5	
Cloud Programming and Software Environments Features of Cloud and Grid Platforms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments.	**08 Hours**

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.

- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. “Distributed and Cloud Computing, From Parallel Processing to the Internet of Things”,Kai Hwang, Geoffrey C. Fox. Jack J Dongarra, MK Publishers, 2012.

Reference Books:

1. “Moving to the Cloud, Developing Apps in the New World of Cloud Computing”, DinakarSitaram, Geetha Manjunath, Elsevier Publication, 2012.
2. “Cloud Computing, A Practical Approach”, Anthony T. Volte, Toby J. Volte and Robert Elsenpeter, McGraw Hill, 2010.
3. “Cloud Computing for Dummies”, J. Hurwitz, ISBN 978-0-470-484-8
4. “Web-Based Applications that Change the Way You Work and Collaborate Online”, Michael Miller, Pearson Publication, 2012.

ARTIFICIAL INTELLIGENCE

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	16MCA543	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcomes (CO): After studying this course, students will be able to:

CO1: Acquire knowledge of

- Uncertainty and Problem solving techniques
- Symbolic knowledge representation to specify domains
- Reasoning tasks of a situated software agent

CO2: Comprehend on

- different logical systems for inference over formal domain representations
- trace on particular inference algorithm working on a given problem specification

CO3: Apply and Analyse AI technique to any given concrete problem

CO4: Interpret and Implement non-trivial AI techniques in a relatively large system

Modules	**Teaching Hours**
Module -1	**08 Hours**
What is Artificial Intelligence: The AI Problems, The Underlying assumption, What is an AI Technique?, The Level of the model, Criteria for success, some general references, One final word and beyond. Problems, problem spaces, and search: Defining, the problem as a state space search, Production systems, Problem characteristics, Production system characteristics,	
Module -2	**08 Hours**
Heuristic search techniques: Generate-and-test, Hill climbing, Best-first search, Problem reduction, Constraint satisfaction, Mean-ends analysis. Knowledge representation issues: Representations and mappings, Approaches to knowledge representation, Issues in knowledge representation, The frame problem.	
Module 3	
Using predicate logic: Representing simple facts in logic, representing instance and ISA relationships, Computable functions and predicates, Resolution, Natural Deduction **Symbolic Reasoning Under Uncertainty**: Introduction to non-monotonic reasoning, Logic for non-monotonic reasoning	**08 Hours**
Module -4	
Implementation: Depth-first search, Implementation: Breadth-first search. Statistical Reasoning: Probability and Bayes Theorem, Certainty factors and rule-based systems, Bayesian Networks, Fuzzy logic.	**08 Hours**
Module -5	
Weak Slot-and-filter structures: Semantic Nets Frames, Strong slot-and –filler structures: Conceptual dependency, scripts, CYC.	**08 Hours**

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Elaine Rich, Kevin Knight, Shivashankar B Nair: Artificial Intelligence, Tata McGraw Hill 3rd edition. 2013

Reference Books:

1. Stuart Russel, Peter Norvig: Artificial Intelligence A Modern Approach, Pearson 3rd edition 2013.
2. Nils J. Nilsson: “Principles of Artificial Intelligence”, Elsevier, ISBN-13: 9780934613101

STORAGE AREA NETWORKS [As per Choice Based Credit System (CBCS) scheme] SEMESTER – V			
Course Code	16MCA544	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03
CREDITS – 03			

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Understand the fundamentals of storage and storage networking concepts
CO2: Analyze Network Attached and Storage Area Networks Requirements
CO3: Apply and Integrate SAN and NAS solutions for an enterprise requirements
CO4: Design a secured, scalable SAN / NAS enterprise solutions

Modules	**Teaching Hours**
Module -1	**8 Hours**
Concepts of Storage Networking The Data Storage and Data Access Problem, The Battle for Size and Access Decoupling the Storage Component: Putting Storage on the Network, Decoupling the Storage Component: Creating a Network for Storage	
Module -2	**8 Hours**
Storage Fundamentals Storage Architectures, Device Overviews, Connectivity Options, Data Organizational Methods	
Module -3	**8 Hours**
Network Attached Storage Putting Storage on the Network, NAS Hardware Devices , NAS Software Components, NAS Connectivity Options	
Module -4	**8 Hours**
Storage Area Networks Architecture Overview, Hardware Devices, Software Components, Configuration Options for SANs.	
Module -5	**8 Hours**
Application Defining the I/O Workload, Applying the SAN Solution, Applying the NAS Solution Considerations When Integrating SAN and NAS **Management** Planning Business Continuity, Managing Availability, Maintaining Serviceability, Capacity Planning and Security Considerations **Case Studies** NAS Case Study, SAN Case Study, SAN/NAS Management Case Study	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. The Complete Reference " Storage networks" , Robert Spalding, McGraw Hill Education (India) 2003

Reference Books:

1. Information Storage and Management (Misl-Wiley) : 2nd Edition, Emc Education Services, Wiley; Second edition (29 August 2012), ISBN-13: 978-8126537501
2. Storage Are networks Essentials : A complete guide to understanding and Implementing SANs, Richard Barker, Paul Massiglia, Wiley

SOFTWARE DEFINED NETWORKS

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	16MCA551	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Recognize the fundamentals and characteristics of Software Defined Networks
CO2: Understand the basics of Software Defined Networks Operations and Data flow
CO3: Discriminate different Software Defined Network Operations and Data Flow
CO4: Analyse alternative definitions of Software Defined Networks
CO5: Apply different Software Defined Network Operations in real world problem

Modules	Teaching Hours
Module -1	**8 Hours**
Introduction to SDN Understanding the SDN, Understanding the SDN technology, Control Plane, Data Plane, Moving information between planes, separation of the control and data planes, Distributed control planes, Load Balancing , Creating the MPLS Overlay, Centralized control planes.	
Module -2	**8 Hours**
Working of SDN Evaluation of Switches and Control planes, SDN Implications, Data centre Needs, Forerunner of SDN ,Software Defines Networks is Born, Sustain SDN interoperability, Open source contribution, Fundamental Characteristics of SDN, SDN Operations, SDN Devices, SDN Controllers, SDN Applications, Alternate SDN methods.	
Module -3	**8 Hours**
The Open Flow Specifications Open Flow Overview, Open Flow Basics, Open Flow 1.0 additions, Open Flow 1.1 additions, Open Flow 1.2 additions, Open Flow 1.3 additions, Open Flow limitations.	
Module -4	**8 Hours**
SDN via APIS, SDN via Hypervisor-Based Overlays, SDN via Opening up the device, Network function virtualization, Alternative Overlap and Ranking.	
Module -5	**8 Hours**
Data centres definition, Data centres demand, tunnelling technologies for Data centres Path technologies in data centres, Ethernet fabrics in Data centres, SDN use case in Data centres	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.

- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Software Defined Networking by Thomas D Nadeau and Ken Gray.
2. Software Define Networks, A Comprehensive Approach, Paul Goransson, Chuck Black. MK Publications.

Reference Books:

1. Software Defined Networking for Dummies brought you by cisco, Brian Underdahl and Gary Kinghorn.

INTERNET OF THINGS (IoT)
[As per Choice Based Credit System (CBCS) scheme]
SEMESTER – V

Subject Code	16MCA552	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Understand constraints and opportunities of wireless and mobile networks for Internet of Things.
CO2: Analyze the societal impact of IoT security events.
CO3: Develop critical thinking skills.
CO4: Analyze, design or develop parts of an Internet of Things solution and map it toward selected business model(s)
CO5: Evaluate ethical and potential security issues related to the Internet of Things.

Modules	**Teaching Hours**
Module -1	**8 Hours**
M2M to IoT Introduction: The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics	
Module -2	**8 Hours**
M2M to IoT- A Market Perspective– Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. **M2M to IoT-An Architectural Overview**– Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.	
Module - 3	**8 Hours**
M2M and IoT Technology Fundamentals Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service(XaaS), M2M and IoT Analytics, Knowledge Management	
Module -4	**8 Hours**
IoT Architecture-State of the Art Introduction, State of the art, **Architecture Reference Model-** Introduction, Reference Model and architecture, IoT reference Model. **IoT Reference Architecture** Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views.	
Module-5	**8 Hours**
Real-World Design Constraints- Introduction, Technical Design constraints-hardware is popular again, Data representation and visualization, Interaction and remote control. **Industrial Automation-** Service-oriented architecture-based device integration, SOCRADES: realizing the enterprise integrated Web of Things, IMC-	

AESOP: from the Web of Things to the Cloud of Things, **Commercial Building Automation-** Introduction, Case study: phase one-commercial building automation today, Case study: phase two- commercial building automation in the future.

Question Paper Pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, StamatisKarnouskos, David Boyle, **"From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence",** 1st Edition, Academic Press, 2014.

Reference Books:

1. Vijay Madisetti and ArshdeepBahga, "Internet of Things (A Hands-on-Approach)", 1st Edition, VPT, 2014.
2. Francis daCosta, "Rethinking the Internet of Things: A Scalable Approach to Connecting Everything", 1st Edition, Apress Publications, 2013

SERVICE ORIENTED ARCHITECTURES (SOA)

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Subject Code	16MCA553	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Describe about evolution, characteristics and services in SOA with SOA architecture, WSDL, SOAP and UDDI.

CO2: Analyze the SOA Architectural style, SOA strategies, modeling web services.

CO3: Design, implementing process of SOA in web service.

CO4: Apply the SOA operational style for the web services.

Modules	**Teaching Hours**
Module -1	**8 Hours**
Service Oriented Architecture Fundamentals of SOA, Common characteristics of contemporary SOA, Common tangible benefits of SOA, A SOA timeline (from XML to Web Services to SOA) **Evolution of SOA** The continuing evolution of SOA (standards organizations and Contributing vendors), The roots of SOA (comparing SOA to Past Architectures)	
Module -2	**8 Hours**
Web Services and Primitives of SOA The Web Services framework, Services (as Web Services), Service Description (with WSDL), Messaging (with SOAP) **Web Services and Contemporary SOA Part 1** Message Exchange patterns, Service Activity; Coordination, Atomic Transactions, Business Activities, Orchestration, Choreography.	
Module -3	**8 Hours**
Web Services and Contemporary SOA Part 2 Addressing, Reliable Messaging, Correlation, Policies, Meta data Exchange, Security, Notification and eventing. **Principles of Service – Orientation** Services- Orientation and the enterprise, Anatomy of service-oriented Architecture, Common Principles of Service Orientation; How Service Orientation principles interrelate, Service Orientation and object orientation, Native Web Service support for service orientation principles.	
Module -4	**8 Hours**
Service Layers Service Orientation and contemporary SOA, Service Layer Abstraction, Application service layer, Business Service Layer, Orchestration Service Layer, Agnostic Services, Service Layer Configuration scenarios.	

Business Process Design WS-BPEL Language basics, WS-Coordination overview, Service oriented business process redesign, WS-Addressing language basics, WS-Reliable messaging language basics.	
Module -5	**8 Hours**
Enterprise Applications Learning Objectives, Architectural Considerations, Solution Architecture for Enterprise Applications, Solution Architecture for Enterprise Applications based on SOA, Software Platforms for Enterprise Applications.	

Question paper pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

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 Khambhapaty, 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
2. Eric Newcomer, Greg Lomow: Understanding SOA with Web Services,Pearson Education, 2009.

SOFTWARE PROJECT MANAGEMENT

[As per Choice Based Credit System (CBCS) scheme]

SEMESTER – V

Course Code	16MCA554	CIE Marks	20
Number of Lecture Hours/Week	03	SEE Marks	80
Total Number of Lecture Hours	40	SEE Hours	03

CREDITS – 03

Course Outcomes (CO):*At the end of this course, the students will be able to*

CO1: Understand the practices and methods for successful software project management

CO2: Identify techniques for requirements, policies and decision making for effective resource management

CO3: Apply the evaluation techniques for estimating cost, benefits, schedule and risk

CO4: Devise a framework for software project management plan for activities, risk, monitoring and control

CO5: Devise a framework to manage people

Modules	**Teaching Hours**
Module -1	**08 Hours**
Introduction to software project management Introduction, Why is Software Project Management important? What is a Project?, Contract Management, Activities Covered by Software Project Management, Plans, Methods and Methodologies, Some ways of categorizing software projects, Stakeholders, Setting Objectives, Business Case, Project Success and Failure, What is Management? Management Control, Traditional versus Modern Project Management Practices	
Module -2	**08 Hours**
Project evaluation & finance Evaluation of Individual Projects, Cost Benefit Evaluation Techniques, Risk Evaluation, Programme Management, Managing allocation of Resources within Programmes, Financial Accounting –An overview – Accounting concepts, Principles & Standards, Ledger posting, Trial balance, Profit and Loss account Balance sheet	
Module -3	**08 Hours**
Activity planning Objectives of Activity Planning, When to Plan, Project Schedules, Sequencing and Scheduling Activities, Network Planning Models, Forward Pass – Backward Pass , Identifying critical path, Activity Float, Shortening Project Duration, Activity on Arrow Networks Risk Management, Nature of Risk, Categories of Risk, A framework for dealing with Risk, Risk Identification, Risk analysis and prioritization, risk planning and risk monitoring	
Module -4	**08 Hours**
Monitoring and control Creating the Framework, Collecting the Data, Review, Project Termination Review, Visualizing Progress, Cost Monitoring, Earned Value Analysis, Prioritizing Monitoring,	

Getting Project Back To Target, Change Control, Software Configuration Management	
Module -5	**08 Hours**
MANAGING PEOPLE AND WORKING IN TEAMS Introduction, Understanding Behavior, Organizational Behavior: A Background, Selecting the Right Person for the Job, Instruction in the Best Methods, Motivation, The Oldham –Hackman Job Characteristics Model, Stress –Health and Safety Working In Teams, Becoming a Team, Decision Making, Leadership	

Question Paper Pattern:

- The question paper will have ten questions.
- Each full question consists of 16 marks.
- There will be 2 full questions (with a maximum of four sub questions) from each module.
- Each full question will have sub questions covering all the topics under a module.
- The students will have to answer 5 full questions, selecting one full question from each module.

Text Books:

1. Bob Hughes, Mike Cotterell, Rajib Mall, “Software Project Management”, Fifth Edition, Tata McGraw Hill, 2011.

2.“Accounting for Management” JawaharLal, 5th Edition, Wheeler Publications, Delhi

Reference Books:

1.JackMarchewka,” Information Technology-Project Management”, Wiley Student Version, 4th Edition, 2013.

2.James P Lewis,”Project Planning, Scheduling & Control”, McGraw Hill, 5th Edition, 2011.

3.PankajJalote,” Software Project Management in Practise”, Pearson Education, 2002