

<b>Advanced C and OOP with C++</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	18MCA11	CIE Marks	40
Number of Lecture Hours/Week	01 Hour Tutorial/ Instructions 03 Hours Theory	SEE Marks	60
Total Number of Lecture Hours	45	SEE Hours	03
CREDITS – 04			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
CO1: Acquire knowledge on advanced C and C++ programming.			
CO2: Analyze the different concepts of C and C++.			
CO3: Design and Develop the solution to a problem using Object Oriented Programming Concepts.			
CO4: Apply OOP's concepts in real world applications.			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module -1</b>			<b>9 Hours</b>
<b>Introduction to C</b> History of C, Importance of C, Features of C, Basic Structure of C Programs. Input and Output: Basic Screen and Keyboard I/O. Control Statements – if, else-if, switch, Control Structures – while, for, do-while. Arrays: Definition, Representation, Single dimension, Two dimensional, Multidimensional Arrays Strings: Introduction to Strings, Operations on strings, String Handling inbuilt functions. Functions: Categories of functions, Call by Value, Call by reference.			
<b>Module -2</b>			<b>9 Hours</b>
<b>Pointers and Structures</b> Pointers, Pointer Expression, Pointer as function arguments, Functions returning pointers, Pointers to functions, Structures and Unions: Declaring and using structure types. <b>Introduction to OOP with C++</b> Classes and Objects: The Origins of C++, What Is Object-Oriented Programming?, Encapsulation, Polymorphism, Inheritance. The General Form of a C++ Program, Friend Functions, Friend Classes, Inline Functions, Constructors and Destructors, Constructors with One Parameter, When Constructors and Destructors Are Executed, Static Class Members, Static Data Members, Static Member Functions, The Scope Resolution Operator, Nested Classes, Passing Objects to Functions, Returning Objects, Object Assignment.			
<b>Module -3</b>			<b>9 Hours</b>
<b>Arrays, Pointers, References, and the Dynamic Allocation Operators</b> Arrays of Objects, Creating Initialized vs. Pointers to Objects, The this Pointer, References, Reference Parameters, Passing References to Objects, Returning References, C++'s Dynamic Allocation Operators new and delete, Initializing Allocated Memory, Allocating Arrays, Allocating Objects. <b>Function Overloading, Copy Constructors and Default Arguments:</b> Function Overloading, Overloading Constructors, Copy Constructors, Default Function Arguments, Function Overloading and Ambiguity. <b>Operator Overloading:</b> Creating a Member Operator Function, Creating Prefix and			

Postfix Forms, of the Increment and decrement Operators, Operator Overloading Restrictions, Operator Overloading Using a Friend Function, Using a Friend to Overload ++ or --, Overloading [ ], Overloading ( ).	
<b>Module -4</b>	<b>9 Hours</b>
<p><b>Inheritance:</b> Base-Class Access Control, Inheritance and protected Members, Protected Base-Class Inheritance, Inheriting Multiple BaseClasses, Constructors, Destructors, and Inheritance, When Constructors and Destructors Are Executed, Passing Parameters to Base-Class Constructors, Granting Access, Virtual Base Classes.</p> <p><b>Virtual Functions and Polymorphism:</b> 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.</p> <p><b>Templates:</b> Generic Functions, A Function with Two GenericTypes, Explicitly Overloading a Generic Function, Overloading a Function Template, Generic Function Restrictions ,Applying Generic Functions, A Generic Sort, Generic Classes, Using Default Arguments with Template Classes, The typename and export Keywords, The Power of Templates .</p>	
<b>Module -5</b>	<b>9 Hours</b>
<p><b>Exception Handling:</b> 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( ), uncaught_exception( ) Function, The exception and bad_exception Classes, Applying Exception Handling.</p> <p><b>Standard C++ I/O Classes :</b> 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 &lt;&lt; and &gt;&gt;, Creating Your Own Inserters, Creating Your Own Extractors, Creating Your Own Manipulator Functions</p> <p><b>C++ File I/O:</b> fstream and the File Classes, Opening and Closing a File, Reading and Writing Text Files, put( ) and get( ), read( ) and write( ), More get( ) Functions, getline( ) , Detecting EOF, peek( ), Obtaining the Current File Position.</p>	
<p><b>Question paper pattern:</b></p> <ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16 marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>	
<p><b>Text Books:</b></p> <ol style="list-style-type: none"> <li>1. <b>Programming in ANSI C, Balaguruswamy</b>, McGraw Hill Education</li> <li>2. Herbert Schildt: C++ The Complete Reference, 4th Edition, Tata McGraw Hill, 2014.</li> <li>3. K R Venugopal, Rajkumar Buyya, TRavishanker: Mastering C++, Tata McGraw Hill.</li> </ol>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Stanley B. Lippmann, Josee Lajore: C++ Primer, 4th Edition, Addison Wesley.</li> <li>2. Object oriented programming with C++, E. Balaguruswamy, Tata McGraw Hill.</li> </ol>	

3. The C Programming Language, Brian W Kernighan, Dennis M Ritchie, PHI, 2nd Edition			
<b>Introduction to UNIX</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
SubjectCode	18MCA12	CIEMarks	40
NumberofLectureHours/Week	01 Hour Tutorial/ Instructions 03 Hours Theory	SEEMarks	60
TotalNumberofLectureHours	45	SEEHours	03
CREDITS–04			
<b>CourseOutcome(CO):Attheendofthiscourse, thestudentswillbeableto</b>			
CO1: UnderstandandexperiencetheUNIXenvironment, Filesystemandhierarchy.			
CO2:Understand the mechanism of process creation and process management using commands.			
CO3:Explorespecial privileged commandsof system administration.			
CO4:Analyzetheusageofdifferentshellcommands,variablesandAWKfiltering.			
CO5: Use Unix commands and language constructs in buildingshellscripts.			
<b>Modules</b>			<b>TeachingHours</b>
<b>Module-1</b>			<b>9Hours</b>
<b>Introductionof UNIXandShell:</b>  Introduction, History, Architecture, Experience the Unix environment, Basiccommandsls,cat, cal,date,calendar,who,printf,tty,sty,uname,passwd,echo, tput,bc,script,spelland ispell, Introductionto ShellScripting,ShellScripts,read, CommandLineArguments,ExitStatusofaCommand,TheLogicalOperators&&and   ,ex it,if, andcaseconditions,expr,sleepandwait,while,until,for,\$,@, redirection, set and shift, trap.			
<b>Module-2</b>			<b>9Hours</b>
<b>UNIXFileSystem:</b>  Thefile, what'sin afilename?Theparent-childrelationship,pwd,theHome directory, absolutepathnames, using absolute pathnames foracommand, cd, mkdir,rmdir,Relativepathnames,TheUNIXfilesystem.BasicFileAttributes:ls- l, the- doption,FilePermissions,chmod,SecurityandFile Permission, users and groups,securitylevel, changingpermission,changingownershipand group,FileAttributes,Morefileattributes:hardlink,symboliclink,umask,find.			
<b>Module-3</b>			
<b>SimpleFilters:</b>  Pr,head,tail,cut,paste,sort,uniq,trcommands,FiltersusingRegularExpression: grep,RegularExpression,egrep, fgrep, sedinstruction, Line Addressing, InsertingandChangingText,Contextaddressing,writingselectedlinestoofile,the- f option,Substitution,PropertiesofRegularExpressions.			

<b>Module-4</b>	<b>9Hours</b>
<p><b>AwkandAdvancedShellProgramming:</b></p> <p>Awk-Advanced Filters: SimpleawkFiltering, Splitting aLineintoFields,printf,the Logical and Relational Operators, Number Processing, Variables, The –f option, BEGINand ENDpositionalParameters,getline,Built-invariables,Arrays,Functions, InterfacewiththeShell,ControlFlow.The shcommand,export, the Command, Conditional Parameter Substitution, Merging Streams, ShellFunctions, eval, ExecStatementandExamples</p>	
<b>Module-5</b>	<b>9Hours</b>
<p><b>ProcessandSystemAdministration:</b></p> <p>Process basics, PS,internal andexternal commands, running jobsinbackground, nice,at and batch,cron,time commands,EssentialSystemAdministrationroot, administrator’sprivileges,startup&amp; shutdown,managingdiskspace,cpio,tar, Customizing the Environment : System Variables, profile, sty, Aliases, CommandHistory,On-line CommandEditing.</p>	
<p><b>Questionpaperpattern:</b></p> <p>Thequestionpaperwillhavetenquestions. Eachfullquestionconsistsof16marks.</p> <p>Therewillbe2fullquestions(withamaximumof foursubquestions)fromeachmodule. Eachfullquestionwillhavesubquestionscoveringallthetopicsundera module.</p>	
<p><b>Text Books:</b></p> <p>1. YourUNIX-TheUltimateGuide, SumitabhaDas, TataMcGrawHill.</p>	
<p><b>ReferenceBooks:</b></p> <p>1. “UnixShellProgramming”, YashwantKanetkar, 2. “BeginningShellScripting”, EricFoster-Johnson, JohnCWelch, MicahAnderson, Wroxpublication. 3. UNIX:Conceptsand Applications, SumitabhaDas, TataMcGrawHill,</p>	

<b>Introduction to Web Programming</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	18MCA13	CIE Marks	40
Number of Lecture Hours/Week	01 Hour Tutorial/ Instructions 03 Hours Theory	SEE Marks	60
Total Number of Lecture Hours	45	SEE Hours	03
CREDITS – 04			
<p><b>Course Outcome (CO): At the end of this course, the students will be able to</b></p> <p>CO1: Understand the fundamentals of web and thereby develop web applications using various web development languages and tools.</p> <p>CO2: Build the ability to select the essential technology needed to develop and implement web Applications.</p> <p>CO3: Use JavaScript and jQuery to develop dynamic and interactive web page.</p> <p>CO4: Write a well formed / valid XML document.</p> <p>CO5: Design XML document with presentation using CSS.</p>			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module -1</b>			<b>9Hours</b>
<p><b>Web Fundamentals</b> Internet, WWW, Web Browsers and Web Servers, URLs, MIME, HTTP, Security, the Web Programmers Toolbox.</p> <p><b>Introduction to XHTML</b> Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links. Lists, Tables, Forms, Frames, syntactic differences between HTML and XHTML.</p>			
<b>Module -2</b>			<b>9Hours</b>
<p><b>Introduction to HTML5</b> New features of HTML5, HTML5 DocType, HTML5 Structure, Tags- nav, section, article, aside, header, footer, HTML5 Form Elements- Search, tel, url, email, number and range, HTML5 Media tags- Audio and video.</p> <p><b>Cascading Style Sheets</b> 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 &lt;span&gt; and &lt;div&gt; tags, Conflict resolution.</p>			
<b>Module -3</b>			<b>9Hours</b>
<p><b>The basics of JavaScript</b> 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.</p> <p><b>JavaScript and XHTML Documents</b> 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.</p>			
<b>Module -4</b>			<b>9Hours</b>
<b>Dynamic Documents with JavaScript</b>			

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	
<b>Introduction to XML</b> Introduction, Syntax of XML, XML Document Structure, Document type definitions, Namespaces, XML schemas, displaying raw XML documents, Displaying XML documents with CSS, Web services.	
<b>Module -5</b>	<b>9Hours</b>
<b>Introduction to jQuery</b> 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.	
<b>Question paper pattern:</b>	
<ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16 marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>	
<b>Text Books:</b>	
<ol style="list-style-type: none"> <li>1. Robert W.Sebesta ,”Programming the World Wide Web”, 4<sup>th</sup>Edition, Pearson education, 2012.</li> <li>2. Kogent Learning solutions Inc., “HTML 5: Covers CSS3, JavaScript, XML, XHTML AJAX, PHP &amp;jQuery: Black Book”, Dreamtech Press.</li> <li>3. Bear Bibeault, Yehuda Katz: jQuery in Action. 3<sup>rd</sup> Edition, DreamTech India,2008.</li> </ol>	
<b>Reference Books:</b>	
<ol style="list-style-type: none"> <li>1. Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 2<sup>nd</sup> Edition, Pearson, 2018.</li> <li>2. Jeffrey C.Jackson: Web Technologies-A Computer Science Perspective, Pearson Education, 7<sup>th</sup> Impression, 2012.</li> <li>3. Chris Bates: Web Programming Building Internet Applications, 3<sup>rd</sup>Edition, WileyIndia, 2009.</li> <li>4. Zak Ruvalcaba Anne Boehm, “Murach's HTML5 and CSS3”, 3<sup>rd</sup> Edition, Murachs/Shroff Publishers &amp; Distributors Pvt Ltd, 2016.</li> </ol>	

<b>Software Engineering</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER –I			
Subject Code	16MCA14	CIE Marks	40
Number of Lecture Hours/Week	03 Hours Theory	SEE Marks	60
Total Number of Lecture Hours	45	SEE Hours	03
CREDITS – 03			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
CO1 :To understand the insight of software development lifecycle and Process			
CO2 : To understand and analyse and design the software Requirements and software design for applications			
CO 3:To design Test cases and apply software Testing Techniques to validate the developed software			
CO4 : To apply metrics and practice the software maintenance Techniques			
Modules			Teaching Hours
<b>Module -1</b>			<b>9 Hours</b>
<b>Introduction</b> Software : Software Characteristics , Software Applications, Software Engineering: A Layered Technology, The Software Process, The Prototyping Model, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Software Process Models			
<b>Module -2</b>			<b>9 Hours</b>
<b>Software Requirement Specifications</b> Requirement Engineering: Value Of A Good SRS, Requirement Process, Requirements Specification, Functional And Non-Functional Requirements, Requirements Engineering Processes, Requirements Elicitation, Requirements Specification, Requirements Validation, Requirements Change:The Elements Of The Analysis Model, Data Modeling, Functional Modeling And Information FlowThe Mechanics Of Structured Analysis: A Case Study On SRS.			
<b>Module -3</b>			<b>9 Hours</b>
<b>Software Design</b> Design: Design Concepts, Function-Oriented Design, Object-Oriented Design,Detailed Design, Design Patterns, Implementation Issues, Verification, And Metrics; A Case Study On Software Design.			
<b>Module -4</b>			<b>9 Hours</b>
<b>Software Testing And Software Metrics</b> Software Testing Fundamentals , Test Case Design , White-Box Testing , Basis Path Testing , Control Structure Testing , Black-Box Testing , A Strategic Approach To Software Testing, Strategic Issues, Integration Testing, Validation Testing, System Testing A Strategic Approach To Software Testing, Strategic Issues, Integration Testing, Validation Testing, System Testing, The Art Of Debugging			
<b>Module -5</b>			<b>9 Hours</b>
<b>Software Metrics And Maintenance</b> Software Quality, A Framework For Technical Software Metrics, Metrics For The			

<p>Analysis Model, Metrics For The Design Model, Metrics For Source Code, Metrics For Testing , Metrics For Maintenance Introduction To The Basic Concepts, The Maintenance Framework, The Maintenance Process</p>	
<p><b>Question paper pattern:</b></p> <ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16 marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>	
<p><b>Text Books:</b></p> <ol style="list-style-type: none"> <li>1. Ian Sommerville : Software Engineering, 9th edition, Person Education Ltd, 2011.</li> <li>2. PankajJalote: Software Engineering, Wiley India Pvt Ltd (2010)</li> </ol>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Roger S Pressman: Software Engineering-A Practitioners approach, 6th Edition, McGraw-Hill, 2010.</li> <li>2. Penny Grubb, Armstrong A. Takang ,<a href="#">Software Maintenance: Concepts And Practice</a>, 2<sup>nd</sup>Edn.</li> </ol>	



<b>Computer Organization and Architecture</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER – I			
Subject Code	18MCA15	CIE Marks	40
Number of Lecture Hours/Week	03 Hours Theory	SEE Marks	60
Total Number of Lecture Hours	45	SEE Hours	03
CREDITS – 03			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
CO1: Understand the basic of Digital Systems			
CO2: Realize the concept of Computer System Organization			
CO3: Apply the concepts of Input/output Organization and Memory System			
CO4: Analyze the Performance of Memory System and Memory Management			
CO5: Analyze and Implementation of Stack Organization			
Modules			Teaching Hours
<b>Module -1</b>			<b>9 Hours</b>
<b>Computer Organizations: -Binary Systems, Combinational Logic and Sequential Logic:</b> Digital computers and Digital systems, Binary numbers, Number base conversion Octal and Hexadecimal numbers, Binary code, Binary storage and registers, Binary logic. Integrated circuits. Digital logic gates. Map Method (2, 3, 4 variable Maps). NAND and NOR implementation, don't care conditions. Introduction to Adders, Subtractors, Decoders, Multiplexers, Sequential logic: Introduction, Different types of Flip Flops, Registers, Shift registers and Ripple counter.			
<b>Module -2</b>			<b>9 Hours</b>
<b>Basics Structure of Computer Hardware and Software:</b> Computer types, Functional units, Basic operational concepts, Bus structure, Software, Performance, Multi processing and Multi computers. Introduction to Assemblers and Compilers. Memory location and Address, Memory operations, Instructions and instructing sequencing, Addressing modes.			
<b>Module -3</b>			<b>9 Hours</b>
<b>Input/Output Organization and Memory Systems:</b> I/O Devices, Interrupts, DMA, Processor example, Buses. Memory system: Basic concepts, Semiconductor RAM memories, Read-Only memories, Speed size and cost, Cache memories, Virtual memories, Secondary memories.			
<b>Module -4</b>			<b>9 Hours</b>
<b>Computer Architecture: - Advances in Memory Systems:</b> Speed Imbalance between the arithmetic and memory units, Advantages of memory hierarchies, Memory interleaving, Problems of managements of memory hierarchies, Operations of Virtual memories. Associated memories. Operations of cache, Comparison of cache and virtual memory systems, Schemes for cache organizations, Word or block replacement, multilevel cache.			
<b>Module -5</b>			<b>9 Hours</b>
<b>General Organization and Control:</b> Addressing schemes: one, two and three address schemes, No address scheme, Address modification and index registers, General purpose registers. Stack			

<p>organization, Use of stack for evaluation of expressions, interrupt processing, Subroutine return, Storing local variables, Storing parameters, Implementation of stacks and stack organized processors.</p>	
<p><b>Question paper pattern:</b></p> <ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16 marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module.</li> </ul>	
<p><b>Text Books:</b></p> <ol style="list-style-type: none"> <li>1. M. Morris Mano, “ Digital Logic and Computer Design”, Pearson, 2012 (Chapters 1.1 - 1.4, 1.6 - 1.9, 2.7, 3.2 - 3.3, 3.6, 3.8, 4.3 - 4.4, 5.5- 5.6, 6.1-6.2 7.2 - 7.4)</li> <li>2. Carl Hamachar and Z. V. S. Zaky, “ Computer Organization”, 5<sup>th</sup> Edition, Tata McGraw Hill (Chapters 1.1 - 1.5, 2.1 - 2.4, 4.1 - 4.4, 5.1 - 5.5, 5.7)</li> <li>3. P. V. S. Rao, “ Computer System Organization and Architecture”, Pearson 2009 (Chapters 8.1 - 8.7 , 10.1 – 10.5)</li> </ol>	

<b>C and C++ LAB</b>			
[As per Choice Based Credit System (CBCS) scheme]			
SEMESTER – I			
Laboratory Code	18MCA16	CIE Marks	40
Number of Hours/Week	02 Hours Laboratory	SEE Marks	60
		SEE Hours	03
CREDITS – 01			
<p><b>Course Outcome (CO): At the end of this course, the students will be able to</b></p> <p>CO1: Learn and implement how to solve common types of computing problems.  CO2: Acquire knowledge on concepts of C and C++ programming.  CO3: Improves the problem solving skills using C and C++.  CO4: Design and develop interactive applications using C++.  CO5: Implement real world problems using oop's.</p>			
<b>Laboratory Experiments:</b>			
1. Write a Menu driven C program to <ol style="list-style-type: none"> <li>a. Accept two numbers n and m</li> <li>b. Sum of all integers ranging from n to m</li> <li>c. Sum of all odd integers ranging from n to m</li> <li>d. Sum of all even integers ranging from n to m</li> <li>e. Display an error message if n &gt; m. Create functions for each of the options.</li> </ol>			
2. Write a menu driven Program in C for the following Array operations <ol style="list-style-type: none"> <li>a. Creating an Array of N Integer Elements</li> <li>b. Display the Array Elements</li> <li>c. Inserting an Element at a given valid Position</li> <li>d. Deleting an Element at a given valid Position</li> <li>e. Exit.</li> </ol>			
3. Write a Program in C for the following String operations (without using built-in functions) <ol style="list-style-type: none"> <li>a. Read a main String (STR), a Pattern String (PAT) and a Replace String (REP)</li> <li>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.</li> </ol>			
4. Define a structure called student having the properties of student_id, student name and branch of the student with a sub structure of marks of 3 subjects. Write a Menu Driven C Program to <ol style="list-style-type: none"> <li>a. Add new student detail</li> <li>b. Delete a student detail</li> <li>c. Display all student details</li> <li>d. Display the name of the student with the best mark</li> <li>e. Display the name of the student with the worst mark</li> <li>f. Display the average marks scored by the students</li> </ol>			
5. Write a Menu driven C function program using pointers to <ol style="list-style-type: none"> <li>a. Read and Initialize 2 pointer variables</li> <li>b. Add 2 pointer variables and return the result using pointer variable.</li> <li>c. Display in-direct value and address of 3 pointer variables.</li> </ol>			

<p>d. Increment the value of 3 pointer variables.</p> <p>e. Display the value and address of local variables and 3 pointer variables.</p>
<p>6. a. Write a C++ program the swap the values of two variables and demonstrates a function using reference(&amp;) operator.</p> <p>b. Write a C++ program the swap the values of two variables and demonstrates a function using pointers.</p>
<p>7. 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.</p>
<p>8. Write a C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a complex number:</p> <p>(i) ADD (a, s2) – where 'a' is an integer (real part) and s2 is a complex number</p> <p>(ii) ADD (s1, s2) – where s1 and s2 are complex numbers.</p> <p>Friend functions and friend classes:</p>
<p>9. 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 &lt;&lt;. If (m1== m2) then m3= m1+m2 and m4 = m1- m2 else display error.</p>
<p>10. 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.</p>
<p>11. I/O streams and functions.</p> <p>Write a program to implement FILE I/O operations on characters. I/O operations include inputting a string, Calculating length of the string, Storing the string in a file, fetching the stored characters from it, etc.</p>
<p>12. Write a program to implement Exception Handling with minimum 2 built in exceptions.</p>
<p style="text-align: center;"><b>PART-B</b> <b>(Mini Project)</b></p> <p><b>Student should carry out menu driven interactive program on any simple real world problem.</b></p> <p><b>Example Scenario: College Student Management System using C and C++</b></p> <p>Apply C++ concepts while implementing the project. Write a class diagram for the complete project with data members and member functions and then develop a menu driven program for COLLEGE:</p> <ul style="list-style-type: none"> <li>• Create C++ classes for College, Student, Faculty, Course and Department.</li> <li>• College is a root abstract base class, Inherit Department and faculty from College, Course from Department, student from department and course.</li> </ul>

- Imposing college timing function through College class by student and faculty classes using pure virtual functions.
- Department can offer many courses and department can have many faculties. Student should enroll for single course.
- Keep the total number of students count using static data member in student class.
- Any number of student objects can be constructed and destructed.
- Calculate marks and average for each student by overloading [].
- Search a student based on usn and write student details to a file.
- Display all student details in highest total marks order.
- Raise an exception for students who have scored less than 40% average.

Students may develop a GUI for the above project using C/C++ Graphics (NOT mandatory).

**Note 1: In the practical Examination each student has to pick one question from a lot of all 12 Questions and demonstrate the Part B Mini Project.**

**Note 2: Project to be carried out with a team of 2 or individual student**

**Note 3: Change of program is not permitted in the Practical Examination.**

<b>UNIX LAB</b>			
[As per Choice Based Credit System (CBCS) scheme]			
SEMESTER – I			
Laboratory Code	18MCA17	CIE Marks	40
Number of Lecture Hours/Week	02 Hours Laboratory	SEE Marks	60
		SEE Hours	03
CREDITS – 01			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
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.			
<b>Laboratory Experiments:</b>			
a) Explore the Unix environment.			
b) Explore vi editor with vimtutor. Perform the following operations using vi editor, but not limited to:			
<ol style="list-style-type: none"> <li>1. Insert character, delete character, replace character</li> <li>2. Save the file and continue working</li> <li>3. Save the file and exit the editor</li> <li>4. Quit the editor</li> <li>5. Quit without saving the file</li> <li>6. Rename a file</li> <li>7. Insert lines, delete lines,</li> <li>8. Set line numbers</li> <li>9. Search for a pattern</li> <li>10. Move forward and backward</li> </ol>			
<b>1a.</b> Write a shell script that takes a valid directory name as an argument and recursively descend all the sub-directories, finds the maximum length of any file in that hierarchy and write this maximum value to the standard output.			
<b>1b.</b> Write a shell script that accepts a path name and creates all the components in that path name as directories. For example, if the script is named mpc, then the command mpc a/b/c/d should create directories a, a/b, a/b/c, a/b/c/d.			
<b>2a.</b> Write a shell script that accepts two filenames as arguments, checks if the permissions for these files are identical and if the permissions are identical, output common permissions and otherwise output each filename followed by its permissions.			
<b>2b.</b> Write a shell script which accepts valid log-in names as arguments and prints their corresponding home directories, if no arguments are specified, print a suitable error message.			

<p><b>3a.</b> Create a script file called file-properties that reads a filename entered and outputs its Properties.</p> <p><b>3b.</b> Write shell script to implement terminal locking (similar to the lock command). It should prompt the user for a password. After accepting the password entered by the user, it must prompt again for the matching password as confirmation and if a match occurs, it must lock the keyword until a matching password is entered again by the user. Note that the script must be written to disregard BREAK, control-D. No time limit need be implemented for the lock duration.</p>
<p><b>4a.</b> Write a shell script that accepts one or more filenames as arguments and converts all of them to uppercase, provided they exist in the current directory.</p> <p><b>4b.</b> Write a shell script that displays all the links to a file specified as the first argument to the script. The second argument, which is optional, can be used to specify in which the search is to begin. If this second argument is not present, the search is to begin in the current working directory. In either case, the starting directory as well as all its subdirectories at all levels must be searched. The script need not include any error checking.</p>
<p><b>5a.</b> Write a shell script that accepts a filename as an argument and displays its creation time if the file exists and if it does not, send an error message.</p> <p><b>5b.</b> Write a shell script to display the calendar for the current month with the current date replaced by *or** depending on whether the date has one digit or two digits.</p>
<p><b>6a.</b> Write a shell script to find a file/s that matches a pattern given as a command line argument in the home directory, display the contents of the file and copy the file into the directory ~/mydir.</p> <p><b>6b.</b> Write a shell script to list all the files in a directory whose filename is at least 10 characters. (use expr command to check the length)</p>
<p><b>7a.</b> Write a shell script that gets executed and displays the message either "Good Morning" or "Good Afternoon" or "Good Evening" depending upon the time at which the user logs in.</p> <p><b>7b.</b> Write a shell script that accepts a list of filenames as its argument, counts and reports the occurrence of each word that is present in the first argument file or other argument files.</p>
<p><b>8a.</b> Write a shell script that determines the period for which the user is working on the system and displays an appropriate message.</p> <p><b>8b.</b> Write a shell script that reports the logging in of a specified user within one minute after he/she logs in. The script automatically terminates if the specified user does not login during the specified period of time.</p>
<p><b>9a.</b> Write a shell script that accepts the filename, starting and ending line numbers as arguments and displays all the lines between the given line numbers.</p> <p><b>9b.</b> Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40<sup>th</sup>, a "\ " is to be appended as the indication of folding and the processing is to be continued with the residue. The input is to be supplied through a textfile created by the user.</p>
<p><b>10a.</b> Write an awk script that accepts a date argument in the form of dd-mm-yy and displays it in</p>

the form month, day and year. The script should check the validity of the argument and in the case of error, display a suitable message.

**10b.** Write an awk script to delete duplicated lines from a text file. The order of the original lines must remain unchanged.

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

Electrical	34
Mechanical	67
Electrical	80
Computer Science	43
Mechanical	65
Civil	98
Computer Science	64

**11b.** Write an awk script to compute gross salary of an employee accordingly to rule given below.

If basic salary is <10000 then HRA=15% of basic & DA=45% of basic

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

**Note 1:** In the practical examination each student has to pick one question from a lot of all 11 questions.

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



<b>WEB PROGRAMMING LAB</b>			
[As per Choice Based Credit System (CBCS) scheme]			
SEMESTER – I			
Laboratory Code	18MCA18	CIE Marks	40
Number of Hours/Week	02 Hours Laboratory	SEE Marks	60
		SEE Hours	03
CREDITS – 01			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
CO1: Develop web pages using HTML and HTML5.			
CO2: Demonstrate the usage of CSS in designing web pages.			
CO3: Execute simple programming questions using JavaScript.			
CO4: Create dynamic web pages by manipulating the DOM elements.			
CO5: Design and implement user interactive dynamic web based applications using jQuery.			
<b>Laboratory Experiments:</b>			
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 (Use of additional tags encouraged).			
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. Develop and demonstrate a XHTML file that includes Javascript script for the following problems:			
a) Input : A number n obtained using prompt Output : The first n Fibonacci numbers			
b) Input : A number n obtained using prompt Output : A table of numbers from 1 to n and their squares using alert			
4. 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. 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.			
<b>Sample output form</b>			

**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	<input type="text"/>
Caramel Popcorn (2 lb. canister)	\$3.50	<input type="text"/>
Caramel Nut Popcorn (2 lb. canister)	\$4.50	<input type="text"/>
Toffee Nut Popcorn (2 lb. canister)	\$5.00	<input type="text"/>

**Payment Method:**

Visa  Master Card  Discover  Check

6. 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.
7. a) Develop and demonstrate, a HTML 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 appealing.  
b)Modify the above program to get the current semester also(restricted to be a number from 1 to 6)
8. Develop and demonstrate a HTML file which includes JavaScript that uses functions for the following problems:  
a. Parameter: A string Output: The position in the string of the left-most vowel.  
b. Parameter: A number Output: The number with its digits in the reverse order.
9. Develop and demonstrate a HTML5 page which contains  
a) Dynamic Progressive bar.  
b) Display Video file using HTML5 video tag.
10. 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 moved from the top stacking position, it returns

to its original position rather than to the bottom.

11. 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 =

Number 2 =

Result =



12. 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, Branch, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.

13. Develop and demonstrate using jQuery to solve the following:
- Limit character input in the text area including count.
  - Based on check box, disable/enable the form submit button.

14. Develop and demonstrate using jQuery to solve the following:
- Fade in and fade out all division elements.
  - Animate an element, by changing its height and width.

**Note 1: In the practical Examination each student has to pick one question from a lot of all 14 questions.**

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

<b>Introduction to Java Programming</b> [As per Choice Based Credit System (CBCS) scheme ] SEMESTER-II			
Subject Code	18MCA21	CIE Marks	40
Number of Lecture Hours/Week	01 Hour Tutorial/ Instructions 03 Hours Theory	SEE Marks	60
Total Number of Lecture	45	SEE Hours	03
CREDITS-04			
<b>Course Outcome(CO):At the end of this course, the students will be able to</b>			
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 runtime polymorphism applications			
CO3: Exemplify the usage of Packages, Interfaces, Exceptions and Multithreading			
CO4: Demonstrate Enumerations, Wrappers, Autoboxing, Generics, collection framework and I/O operations			
CO5: Implement the concepts of Networking using Java network classes			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module-1</b>			<b>9 Hours</b>
<p><b>Java Programming Fundamentals</b> The Java Language, The Key Attributes of Object-Oriented Programming, The Java Development Kit, A First Simple Program, Handling Syntax Errors, The Java Keywords, Identifiers in Java, The Java Class Libraries.</p> <p><b>Introducing Data Types and Operators</b> 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.</p> <p><b>Program Control Statements</b> 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.</p> <p><b>Introduction to Classes, Objects and Methods</b> 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.</p> <p><b>More Data Types and Operators</b> Arrays, Multidimensional Arrays, Alternative Array Declaration Syntax, Assigning Array References, Using the Length Member, The For-Each Style for Loop, Strings, The Bitwise operators.</p>			

<b>Module-2</b>	<b>9 Hours</b>
<p><b>String Handling</b> String Fundamentals, The String Constructors, Three String-Related Language Features, The Length() Method, Obtaining the characters within a string, String comparison, using indexOf() and last IndexOf(), Changing the case of characters within a string, String Buffer and String Builder.</p> <p><b>A Closer Look at Methods and Classes</b> 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.</p>	
<b>Module-3</b>	<b>9 Hours</b>
<p><b>Inheritance</b> Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call Super class constructors, Using super to Access Super class 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.</p> <p><b>Interfaces</b> Interface Fundamentals, Creating an Interface, Implementing an Interface, Using Interface References, Implementing Multiple Interfaces, Constants in Interfaces, Interfaces can be extended, Nested Interfaces</p> <p><b>Packages</b> Package Fundamentals, Packages and Member Access, Importing Packages, Static Import</p>	
<b>Module-4</b>	<b>9 Hours</b>
<p><b>Exception Handling</b> 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 JDK8, Creating Exception Subclasses.</p> <p><b>Multithreaded Programming</b> 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 notifyAll(),</p>	

Module-5	9 Hours
<p><b>Enumerations, Autoboxing and Annotations</b> Enumerations, Java Enumeration are class types, TheValues() and Valueof() Methods, Constructors, methods, instance variables and enumerations, Autoboxing, Annotations(metadata)</p> <p><b>Networking with Java.net</b> Networking fundamentals, The Networking classes and Interfaces, The Inet Address class, The Socket Class, The URL class, The URL Connection Class, The Http URL Connection Class.</p> <p><b>Exploring Collection Framework</b> Collections Overview, The Collection Interfaces, The collection Classes. The Arrays Class.</p>	
<p><b>Question paper pattern:</b></p> <ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module</li> </ul>	
<p><b>Text Books:</b></p> <ol style="list-style-type: none"> <li>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)</li> </ol>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Java Programming by Hari Mohan Pandey, Pearson Education, 2012.</li> <li>2. Java 6 Programming, Black Book, KoGenT, Dreamtech Press, 2012.</li> <li>3. Java 2 Essentials, Cay Hortsman, second edition, Wiley</li> </ol>	

<b>Data Structures using C++</b> [As per Choice Based Credit System (CBCS) scheme ] SEMESTER–II			
Subject Code	18MCA22	CIE Marks	40
Number of Lecture Hours/Week	01 Hour Tutorial/ Instructions 03 Hours Theory	SEE Marks	60
Total Number of Lecture Hours	45	SEE Hours	03
CREDITS–04			
<b>Course Outcome(CO):At the end of this course, the students will be able to</b> CO1: Acquire knowledge of - Various types of data structures, operations and algorithms - Sorting and searching operations CO2: Analyze the performance of - Stack, Queue, Lists, Trees, Hashing, Searching and Sorting techniques CO3: Implement all the applications of Data structures in a high-level language CO4: Design and apply appropriate data structures for solving computing problems			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module-1</b>			<b>9 Hours</b>
<b>Classification of Data Structures:</b> Primitive and Non- Primitive, Linear and Nonlinear; Data structure Operations, ADT, Array as ADT, Operations - Insert, Delete, Search, Sort, <b>String</b> Definition, Representation, String as ADT, Operations – Insert, Delete, Concatenate, Comparing, Substring. <b>Stack:</b> 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;			
<b>Module-2</b>			<b>9 Hours</b>
<b>Recursion</b> - Recursive definition and processes, Properties of recursive definition or Algorithm, Recursive algorithms: Factorial, GCD, Fibonacci Sequence, Tower of Hanoi <b>Queue:</b> Definition, Representation, Queue as ADT, Operations, Queue Variants: Circular Queue, Priority Queue, Double Ended Queue; Applications of Queues. Programming Examples.			
<b>Module-3</b>			<b>9 Hours</b>
<b>Linked List:</b> Limitations of array implementation, Memory Management: Static (Stack) and Dynamic (Heap) Memory Allocation, Memory management functions. Definition, Representation, Operations: getnode() and Freenode() operations, Types: <b>Singly Linked List</b> . 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.: <b>Circular Linked List:</b> Inserting, deleting and searching elements in a lists, <b>Double Linked List:</b> Inserting and Deleting Nodes, Queue as doubly linked lists, such as insert into position, Delete an specified element. <b>Application of Linked Lists:</b> Stacks, Queues, Double-ended Queues, Priority Queues, Sparse Matrix and Polynomials using Lists, Trees, BST.			

<b>Module-4</b>	<b>9 Hours</b>
<p><b>Trees:</b> 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.</p>	
<b>Module-5</b>	<b>9 Hours</b>
<p><b>Sorting &amp; Searching:</b> Bubble sort, Insertion Sort, Selection sort, Quick sort, Linear Search, Binary Search and BST.</p> <p><b>Hashing:</b> The Hash Table organizations, Hashing Functions, Static and Dynamic Hashing, Collision-Resolution Techniques, Programming Examples.</p>	
<p><b>Question paper pattern:</b></p> <ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module</li> </ul>	
<p><b>Text Books:</b></p> <ol style="list-style-type: none"> <li>2. <b>Programming in ANSI C, Balaguruswamy</b>, McGraw Hill Education</li> <li>3. <b>Data Structures Using C and C++</b> by YedidyahLangsam and Moshe J. Augenstein and Aaron M Tenanbanum, 2nd Edition, Pearson Education Asia, 2002.</li> <li>4. <b>Introduction to Data Structure and Algorithms with C++</b> by Glenn W. Rowe</li> </ol>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. <b>Principles of Data Structures using C &amp; C++</b> by Vinu V. Das, New Age International, 2006</li> <li>2. <b>Data Structures Using C , Balaguruswamy</b>;, McGraw Hill Education</li> </ol>	



<b>MATHEMATICAL STRUCTURS AND STATISTICS</b>			
<b>[As perChoiceBasedCreditSystem (CBCS)scheme]</b>			
<b>SEMESTER-II</b>			
SubjectCode	18MCA23	CIEMarks	40
NumberofLectureHours/Week	01 Hour Tutorial/ Instructions 03 Hours Theory	SEEMarks	60
TotalNumberofLectureHours	45	SEEHours	03
CREDITS - 04			
<b>Courseoutcomes:</b>			
<b><i>CourseOutcome(CO):Attheendofthiscourse, thestudentswill beableto</i></b>			
CO1:Usethelogicalnotationto defineandreasonaboutfundamental mathematicalconceptssuchassets,relations,functions,andintegers.			
CO2:Calculatenumbersof possibleoutcomesof elementarycombinatorial processessuchaspermutationsandcombinations.			
CO3: Calculate probabilities and distributions for simplecombinatorialprocesses;calculateexpectations.			
CO4: Apply statistical methods for correlation and regression. Fitting a curve to a discrete data.			
<b>Modules</b>			<b>Teaching</b>
<b>Module-1</b>			
<b>Fundamentals of Logic:</b> Basic connectives and truth tables, logical equivalence, laws of logic, logical implication rules of inference. Quantifiers			<b>9Hours</b>
<b>Module-2</b>			
<b>Sets Theory and Probability</b> Sets and subsets,set operations, laws of set thery, counting and venn diagrams. A first word on probability, axioms of probability, conditional probability, Bayes' theorem.			<b>9Hours</b>
<b>Module-3</b>			
<b>Fundamentals of Counting and Properties of Integers:</b> The rules of Sum and Product, Permutations and Combinations, The Binomial theorem, Mathematical Induction, Recursive definitions: Fibonacci and Lucas numbers			<b>9Hours</b>

<b>Module-4</b>	<b>9Hours</b>
<p><b>Random variables and Probability Distributions:</b></p> <p>Concept of a random variable Discrete probability distributions, Continuous probability distributions, Mean, Variance and Covariance of random variables. Binomial and Poisson distributions, Exponential and Normal distributions with mean, variables and problems.</p>	
<b>Module-5</b>	<b>9 Hours</b>
<p><b>Statistical methods and Curve Fitting:</b></p> <p>Correlation, coefficient of correlations, lines of regression-principle of least square. Rank correlation. Curve Fitting- Graphical method, Principle of least square- to fit a straight line and parabola. Fitting of other curves of the form  <math>y = ax^b</math> <math>y = ae^{bx}</math> <math>xy^n = b</math></p>	
<p><b>Question paper pattern:</b></p> <ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> </ul> <p>The students will have to answer 5 full questions, selecting one full question from each module</p>	
<p><b>Text Books:</b></p> <ol style="list-style-type: none"> <li>1. Discrete and Combinatorial Mathematics by Ralph P. Grimaldi and B. V. Ramana, 5<sup>th</sup> edition, Pearson, 2011. (Chapters: 1.1 to 1.3, 4.1, 4.2, 2.1 to 2.5, 3.1 to 3.6)</li> <li>2. Probability and Statistics for Engineers and Scientists by Walpole, Myers, Myers, Ye, Pearson Education, Eighth edition. (Chapters: 3.1–3.3, 4.1 to 4.4, 5.3, 5.6, 6.2 to 6.4, 6.6, 6.7)</li> <li>3. Higher Engineering Mathematics by Dr. B. S. Grewal, Khanna publishers, 40<sup>th</sup> edition (Chapters: 25.12 to 25.16, 24.1 to 24.6)</li> </ol>	

**ReferenceBooks:**

1. Discrete Mathematics and its Applications by Kenneth H Rosen, 7<sup>th</sup> edition, (Indian adaptation by Kamala Krithivasan), Tata McGraw Hill, 2011.
2. Discrete Mathematical Structures with Applications to Computer Science by J. P. Tremblay and R. Manohar, McGraw Hill.
3. Probability and Statistics for Engineers by Richard A. Johnson and C. B. Gupta, Pearson Education.
4. Discrete Mathematics by J. K. Sharma, Macmillan Publishers India Ltd. 3<sup>rd</sup> edition 2011.

<b>Computer Networks</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER –II			
Subject Code	18MCA24	CIE Marks	40
Number of Lecture Hours/Week	03	SEE Marks	60
Total Number of Lecture Hours	45	SEE Hours	03
CREDITS – 03			
<b>Course Outcome (CO): The students will be able to</b>			
CO1: Understand the computer network concepts.			
CO2: Know various types of Networks & Communication medias			
CO3: Identify the components required to build different types of networks			
CO4: understand layering concepts of TCP/IP and OSI models			
CO5: Understand the working principles of various application protocols			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module -1</b>			<b>9 Hours</b>
<b>Introduction to Computer Networks and Physical Layer</b> Networking Devices, Classification of Computer Networks, Network Protocol Stack (TCP/IP and ISO-OSI), Data Transmission Concepts, Analog and Digital Data Transmission, Communication media, Digital modulation techniques (FDMA,TDMA,CDMA).			
<b>Module -2</b>			<b>9 Hours</b>
<b>Data Link layer</b> Data link layer design issues, Error Detection and Correction Codes, Data Link Protocols - Stop-and-wait protocol, Go-back-n protocol, selective repeat protocol. Wireless LAN, Bluetooth.			
<b>Module -3</b>			<b>9 Hours</b>
<b>Network Layer</b> Network Layer Design issues, Routing algorithms, Congestion Control Algorithms, Internetworking			
<b>Module -4</b>			<b>9 Hours</b>
<b>The Transport Layer</b> The Transport Service, Elements of Transport Protocols, The Internet Transport Protocol: UDP, The Internet Transport Protocols – TCP.			
<b>Module -5</b>			<b>9 Hours</b>
<b>The application Layer</b> 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.			
<b>Question paper pattern:</b>			
<ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16 marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module</li> </ul>			

**Text Books:**

1. "Computer Networks" by Andrew S Tanenbaum, David J Wetheral, 5th Edition, Pearson 2012
- 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

<b>Operating System</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER –II			
Subject Code	18MCA25	CIE Marks	40
Number of Lecture Hours/Week	03	SEE Marks	60
Total Number of Lecture Hours	45	SEE Hours	03
CREDITS – 03			
<p><b>Course Outcome (CO): The students will be able to</b></p> <p>CO1: Understand the basic of Computer and Operating System Structure</p> <p>CO2: Realize the concept of Process Management</p> <p>CO3: Analyze and Evaluation of Synchronization and Deadlock</p> <p>CO4: Understand the concepts of System Software</p> <p>CO5: Design and Implementation of Assemblers, Loaders and Linkers</p>			
Modules			Teaching Hours
<b>Module -1</b>			<b>9 Hours</b>
<p><b>Computer and Operating System Structure:</b> Introduction to operating system, Computer system architecture, Operating system structure, Operating system services, System call, Types of system call, System programs, System structure, Virtual machines, Operating system generation.</p>			
<b>Module -2</b>			<b>9 Hours</b>
<p><b>Process Management:</b> Process Concept, Process scheduling, Inter process communication, Multithreaded programming, Multithreading models, Process scheduling: Basic concept, Scheduling criteria, Scheduling algorithms.</p>			
<b>Module -3</b>			<b>9 Hours</b>
<p><b>Synchronization, Deadlock and Memory management:</b> Critical section problem, Petersons solution, Synchronization hardware, Semaphores, Classic problems of synchronization, Monitors, System model, Deadlock characterization, Deadlock avoidance, Deadlock detection, Memory management strategies: Swapping, Contiguous memory allocation, paging, page. replacement.</p>			
<b>Module -4</b>			<b>9 Hours</b>
<p><b>Machine Architecture and Introduction to Assembles:</b> Introduction, System software and Machine architecture. Simplified Instructional Computer (SIC): SIC Machines architecture, SIC/XE Machine architecture, SIC Programming examples. Traditional machines (CISC): VAX architecture. RISC machines: UltraSPARC architecture. Basic assembler functions: A simple SIC assembler, assembler algorithm and data structures. Machine dependent assembler features: Instructional formats and addressing modes, program relocation.</p>			

Module -5	9 Hours
<p><b>Assemblers and Loaders:</b>  Machine independent assembler features: Literals, symbol definition statements, expression, program blocks, control sections and program linking. Assembler design options: One pass assemblers, multi pass assemblers. 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</p>	
<p><b>Question paper pattern:</b></p> <ul style="list-style-type: none"> <li>• The question paper will have ten questions.</li> <li>• Each full question consists of 16 marks.</li> <li>• There will be 2 full questions (with a maximum of four sub questions) from each module.</li> <li>• Each full question will have sub questions covering all the topics under a module.</li> <li>• The students will have to answer 5 full questions, selecting one full question from each module</li> </ul>	
<p><b>Text Books:</b></p> <ol style="list-style-type: none"> <li>1. A. Silberschatz, P. B. Galvin and G. Gagne: "Operating System Concept" 8th edition Wiley India Edition, Reprint 2012. (Chapters: 1.1 - 1.4, 2.1, 2.3, 2.4, 2.5, 2.7, 2.8, 2.10, 3.1, 3.2, 3.4, 4.2, 5.1 -5.3, 6.2 - 6.7, 7.1, 7.2, 7.5, 7.6, 8.2 - 8.4, 9.4)</li> <li>2. Leland L. Beck: System Software, 3<sup>rd</sup> Edition, Addison Wesley,1997. (Chapters: 1.1, 1.2, 1.3, 1.4, 1.5, 2.1, 2.2, 2.3, 2.4, 3.1, 3.2, 3.3)</li> </ol>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Chakraborty: "Operating Systems" Jaico publishing house,2011</li> <li>2. Dhananjaya M. Dhamdhere: "Operating Systems – A concept – Based approach", Tata McGraw Hill, 3<sup>rd</sup> Edition 2012</li> <li>3. J. Nithyashri, "System Software", 2<sup>nd</sup> Edition, Tata McGraw Hill. 2010</li> </ol>	

<b>Java Programming Lab</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER –II			
Laboratory Code	18MCA26	CIE Marks	40
Number of Hours/Week	02 Hours Laboratory	SEE Marks	60
		SEE Hours	03
CREDITS – 01			
<b>Course Outcome (CO): The students will be able to</b>			
CO1: Understand Java 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.			
<b>PART A</b>			
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 String Buffer 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 Stack operations that demonstrates the runtime binding.

10. Write a JAVA program which uses FileInputStream / FileOutputStream Classes.

11. Write JAVA programs which demonstrate utilities of Linked List Class.

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

**PART B**

**(Mini Project)**

**Student should carryout menu driven interactive program on any simple real world problem using JAVA.**

**Students may do the above project with GUI as well.**

**Note 1: In the practical Examination student has to execute one program from a lot of all the 12 questions and demonstrate Part B Mini Project.**

**Note 2: Project to be carried out with a team of 2 or individual student**

**Note 3: Change of program is not permitted in the Practical Examination.**

<b>Data Structures Lab</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER –II			
Laboratory Code	18MCA27	CIE Marks	40
Number of Hours/Week	02 Hours Laboratory	SEE Marks	60
		SEE Hours	03
CREDITS – 01			
<b>Course Outcome (CO): The students will be able to</b>			
CO1: Acquire knowledge of <ul style="list-style-type: none"> <li>- Various types of data structures, operations and algorithms</li> <li>- Sorting and searching operations</li> </ul>			
CO2: Analyze the performance of <ul style="list-style-type: none"> <li>- Stack, Queue, Lists, Trees, Hashing, Searching and Sorting techniques</li> </ul>			
CO3: Implement all the applications of Data structures in a high-level language			
CO4: Design and apply appropriate data structures for solving computing problems			
1. Convert a prefix notation to postfix notation.			
2. Evaluate a given postfix expression and its values for the variables.			
3. Simulate the working of circular queue providing the following operations–Insert, Delete and Display.			
4. Demonstrate recursion <ol style="list-style-type: none"> <li>a. Calculate GCD and LCM of 3 integer numbers</li> <li>b. Solve Towers of Hanoi Problem</li> <li>c. Calculate the sum for a given number ‘n’ from 1 to n.</li> </ol>			
5. Simulate the working of a linked list providing the following operations <ol style="list-style-type: none"> <li>a. Insert at the beginning</li> <li>b. Insert at the end</li> <li>c. Insert before a given element</li> <li>d. Insert at the position</li> <li>e. Display</li> </ol>			
6. Simulate the working of a circular linked list providing the following operations <ol style="list-style-type: none"> <li>a. Delete from the beginning</li> <li>b. Delete from the end</li> <li>c. Delete a given element</li> <li>d. Delete every alternate element</li> <li>e. Display</li> </ol>			
7. Simulate the working of a dequeue.			
8. Simulate the working of a double linked list to implement stack and queue.			
9. Create a binary tree and implement the tree traversal techniques of inorder, preorder and postorder.			
10. Implement quick sort.			
11. Implement Heap sort.			

12. Implement the search techniques of

- a. Linear Search
- b. Binary Search

13 Write a Program to

- a) Create AVL Tree
- b) Insert element to AVL tree
- c) Find the height of the AVL tree

**Note 1: In the practical Examination student has to execute one program from a lot of all the 13 questions.**

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

<b>Mathematics and Statistics Lab</b> [As per Choice Based Credit System (CBCS) scheme] SEMESTER –II			
Laboratory Code	18MCA28	CIE Marks	40
Number of Hours/Week	02 Hours Laboratory	SEE Marks	60
		SEE Hours	03
CREDITS – 01			
Problems based on Mathematics and Statistics will be formulated in due course			