

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

**(A Unit of Nitte Education Trust (R), Mangalore)**

**An Autonomous Institution**

**Department of Master of Computer Applications**

**Syllabus for Master of Computer  
Applications (M.C.A)**

## VISION

- Empowerment through **Quality Education and Technical Competency** with a focus on Computer Applications for development of self and society

## MISSION

- To provide quality and industry-oriented education in applied Computer Science with conceptual understanding and practical perception
- To prepare students for exciting global careers in the ever-changing IT industry
- To provide holistic development of students

## PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

### [career and professional achievements]

- To develop the students' technical skills, analytical skills and professional skills to design and develop computer applications to meet the customer end IT industry and also adapt to every changing IT technologies
- To develop the students' communication skills for effective delivery of a team composed of different region and disciplines in order to be an effective and efficient leader of the team in IT/ITES industry
- To develop the students' social, ethical, integrity so that it can reflect its impact on society through their profession and behavior.

## GRADUATE ATTRIBUTES

### (Defined by NBA)

1. **Computational Knowledge:** Apply knowledge of computing fundamentals, computing specialization, mathematics and domain knowledge appropriate for the computing specialization to the abstraction and conceptualization of computing models from defined problems and requirements.
2. **Problem Analysis:** Identify, formulate, research literature and solve complex computing problems reaching substantiated conclusions using fundamental principles of Mathematics, Computing Sciences and relevant domain disciplines.
3. **Design / Development of solutions:** Design and evaluate solutions for complex computing problems and evaluate systems, components or processes that meet specified needs with appropriate considerations for public health and safety, cultural societal and environmental considerations.

4. **Conduct Investigations of complex computing problems:** Use research based knowledge and research methods including design of experiments, analysis and interpretation of data and synthesis of the information to provide valid conclusions.
5. **Modern Tool Usage:** Create, select, adopt and apply appropriate techniques, resources and modern computing tools to complex computing activities with an understanding of the limitations.
6. **Professional Ethics:** Understand and commit to professional ethics and cyber regulations, responsibilities and norms of professional computing practice.
7. **Life Long Learning:** Recognize the need and have the ability to engage in independent learning for continual development as a Computing Professional.
8. **Project Management and Finance:** Demonstrate knowledge and understanding of the computing and management principles and apply these to one's own work as a member and leader in a team to manage projects and in multidisciplinary environments.
9. **Communication Efficacy:** Communicate effectively with the computing community and society at large about complex computing activities by being able to comprehend and write effective reports and design documentation, make effective presentations and give and understand clear instructions.
10. **Societal and environmental concern:** Understand and assess societal, environmental, health safety, legal and cultural issues within local and global contexts and consequential responsibilities relevant to professional computing practice.
11. **Individual and Team work:** Function effectively as an individual and as a member or leader in diverse teams in multi-disciplinary environments.
12. **Innovation and entrepreneurship:** Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large.

#### **PROGRAM OUTCOMES (POs)**

- A. To equip students with core computing principles, technical, analytical and managerial abilities to compete in a global environment.
- B. To equip students with the ability to analyze and assess problems to meet users requirements.
- C. To equip students with an ability to apply design and development principles in creating software systems for diverse customers.
- D. To equip students with the ability to use research based knowledge and research methods for interpretation of data and synthesis of the information to provide valid conclusions.
- E. To equip students with the ability to use current techniques, skills and tools for computing practices.
- F. Mould the students to understand the impact of professional and ethical IT solutions in societal, environmental context and of sustainable development.

- G. Students will have the ability for self-improvement through continuous professional development and life-long learning.
- H. Students will have the ability to employ effective project management skills to plan and develop projects.
- I. Students will have an ability to communicate effectively and develop leadership qualities and interpersonal skills.
- J. To equip students with an ability to analyze and assess societal, safety, legal and cultural issues and the consequent responsibilities to the professional practice.
- K. To equip students with an ability to function effectively as an individual and as a member of diverse teams in multi-disciplinary environments.
- L. To equip students with the ability to become innovative leaders and entrepreneurs and pursue opportunity to create value and wealth for betterment of self and society.

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF M.C.A**  
**SCHEME OF TEACHING AND EXAMINATION 2016-2019**

<b>I SEMESTER</b>					<b>CREDIT BASED</b>			
<b>Subject Code</b>	<b>Name of the Subject</b>	<b>Teaching hours/week</b>		<b>Duration of Exam in Hours</b>	<b>Marks for</b>		<b>Total marks</b>	<b>Credits</b>
		<b>Lecture</b>	<b>Practical/Assignment</b>		<b>CI</b>	<b>SEE</b>		
16MCA101	Problem solving using C	4	-	3	50	50	100	4
16MCA102	UNIX Programming	4	-	3	50	50	100	4
16MCA103	Web Technologies	4	-	3	50	50	100	4
16MCA104	Computer Organization	4	-	3	50	50	100	4
16MCA105	Professional Communication and Management	4	-	3	50	50	100	4
16MCA106 L	C Laboratory	-	3	3	50	50	100	1.5
16MCA107 L	UNIX Laboratory	-	3	3	50	50	100	1.5
16MCA108 L	Web Designing Laboratory	-	3	3	50	50	100	1.5
<b>Total</b>		<b>20</b>	<b>9</b>	<b>24</b>	<b>400</b>	<b>400</b>	<b>800</b>	<b>24.5</b>

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF M.C.A**  
**SCHEME OF TEACHING AND EXAMINATION 2016-2019**

<b>II SEMESTER</b>					<b>CREDIT BASED</b>			
<b>Subject Code</b>	<b>Name of the Subject</b>	<b>Teaching hours/week</b>		<b>Duration of Exam in Hours</b>	<b>Marks for</b>		<b>Total marks</b>	<b>Credits</b>
		<b>Lecture</b>	<b>Practical/ Assignment</b>		<b>CIE</b>	<b>SEE</b>		
16MCA201	Object Oriented Programming with C++	4	-	3	50	50	100	4
16MCA202	Data Structures using C	4	-	3	50	50	100	4
16MCA203	Database Management System	4	-	3	50	50	100	4
16MCA204	Operating System	4	-	3	50	50	100	4
16MCA205	Systems Software	4	-	3	50	50	100	4
16MCA206L	Object Oriented Programming with C++ Laboratory	-	3	3	50	50	100	1.5
16MCA207L	Data Structures Laboratory	-	3	3	50	50	100	1.5
16MCA208L	Database Management System Laboratory	-	3	3	50	50	100	1.5
16MCA209S	Technical Seminar	-	2	-	50	50	100	1
<b>Total</b>		<b>20</b>	<b>11</b>	<b>24</b>	<b>450</b>	<b>450</b>	<b>900</b>	<b>25.5</b>

**16MCA209S Seminar**

- Students should present the seminar on cutting edge/emerging/state of the art technologies in the field of Computer Science and Applications.
- Duration of the seminar should be approximately 45 minutes.
- Student should submit the write up on seminar topic containing at least 10 pages

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF M.C.A**  
**SCHEME OF TEACHING AND EXAMINATION 2016-2019**

<b>III SEMESTER</b>					<b>CREDIT BASED</b>			
<b>Subject Code</b>	<b>Name of the Subject</b>	<b>Teaching hours/week</b>		<b>Duration of Exam in Hours</b>	<b>Marks for</b>		<b>Total marks</b>	<b>Credits</b>
		<b>Lecture</b>	<b>Practical/ Assignment</b>		<b>CIE</b>	<b>SEE</b>		
16MCA301	Computer Networks	4	-	3	50	50	100	4
16MCA302	Java Programming	4	-	3	50	50	100	4
16MCA303	Python Programming	4	-	3	50	50	100	4
16MCA304	Discrete Mathematical Structures	4	-	3	50	50	100	4
16MCA305	Software Engineering	3	-	3	50	50	100	3
16MCA306E	Elective- 1	3	-	3	50	50	100	3
16MCA307L	Java Programming Laboratory	-	3	3	50	50	100	1.5
16MCA308L	Python Laboratory	-	3	3	50	50	100	1.5
<b>Total</b>		<b>22</b>	<b>6</b>	<b>24</b>	<b>400</b>	<b>400</b>	<b>800</b>	<b>25</b>
<b>Elective – 1</b>								
16MCA3061E		Intellectual Property Rights						
16MCA3062E		Enterprise Resource Planning(ERP)						
16MCA3063E		MIS & E-Commerce						
16MCA3064E		Cyber Security						

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF M.C.A**  
**SCHEME OF TEACHING AND EXAMINATION 2016-2019**

<b>IV SEMESTER</b>				<b>CREDIT BASED</b>				
<b>Subject Code</b>	<b>Name of the Subject</b>	<b>Teaching hours/week</b>		<b>Duration of Exam in Hours</b>	<b>Marks for</b>		<b>Total marks</b>	<b>Credits</b>
		<b>Lecture</b>	<b>Practical/Assignment</b>		<b>CI</b>	<b>SE</b>		
16MCA401	Advanced Java Programming	4	-	3	50	50	100	4
16MCA402	Analysis and Design of Algorithm	4	-	3	50	50	100	4
16MCA403	Software Testing and Practices	4	-	3	50	50	100	4
16MCA404	Principles of User Interface Design	3	-	3	50	50	100	3
16MCA405 E	Elective -2	3	-	3	50	50	100	3
16MCA406 E	Elective -3	3	-	3	50	50	100	3
16MCA407 L	Advanced Java Programming Laboratory	-	3	3	50	50	100	1.5
16MCA408 L	Software Testing Laboratory	-	3	3	50	50	100	1.5
16MCA409 S	Seminar(web Based)	-	2	-	50	50	100	1
<b>Total</b>		<b>21</b>	<b>8</b>	<b>24</b>	<b>450</b>	<b>450</b>	<b>900</b>	<b>25</b>



<b>Elective - 2</b>		<b>Elective -3</b>	
16MCA4051E	Advanced Computer Networks	16MCA4061E	Mobile Technologies
16MCA4052E	Data Mining	16MCA4062E	Big Data Analytics
16MCA4053E	Software Architecture	16MCA4063E	Software Quality Management
16MCA4054E	Cryptography and Network Security	16MCA4064E	Information Retrieval and Search Engines

**16MCA409S Seminar:**

- Duration of the seminar should be approximately 45 minutes.
- Student should submit the write up on seminar topic containing at least 10 pages

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF M.C.A**  
**SCHEME OF TEACHING AND EXAMINATION 2016-2019**

V SEMESTER					CREDIT BASED			
Subject Code	Name of the Subject	Teaching hours/week		Duration of Exam in Hours	Marks for		Total marks	Credits
		Lecture	Practical/Assignment		CIE	SEE		
16MCA51	Object Oriented Modelling and Design Patterns	4	2	3	50	50	100	5
16MCA52	Programming Using C# and .NET	4	-	3	50	50	100	4
16MCA53	Mobile Applications	4	-	3	50	50	100	4
16MCA54	Cloud Computing	3		3	50	50	100	3
16MCA55	Elective - 4	3	-	3	50	50	100	3
16MCA56	Elective -5	3	-	3	50	50	100	3
16MCA57	.NET Laboratory	-	3	3	50	50	100	1.5
16MCA58	Mini Project: Mobile Applications in Android	-	3	3	50	50	100	1.5
<b>Total</b>		<b>21</b>	<b>8</b>	<b>24</b>	<b>400</b>	<b>400</b>	<b>800</b>	<b>25</b>
<b>Elective - 4</b>				<b>Elective -5</b>				
16MCA551	Web 2.0 and Rich Internet Applications			16MCA561			Software Defined Networks	
16MCA552	Storage Area Networks			16MCA562			Internet Of Things(IOT)	
16MCA553	Artificial Intelligence			16MCA563			Service Oriented Architecture	
16MCA554	Multimedia			16MCA564			Software Project Management	

**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**  
**DEPARTMENT OF M.C.A**  
**SCHEME OF TEACHING AND EXAMINATION 2016-2019**

<b>VI SEMESTER</b>		<b>CREDIT BASED</b>			
<b>Subject Code</b>	<b>Subject Name</b>	<b>Examination</b>			<b>Credits</b>
		<b>CIE</b>	<b>SEE</b>	<b>Total</b>	
16MCA601P	Dissertation Assessment	75	75	150	10
16MCA602P	Internal Evaluation and Viva voce	75	75	150	15
	<b>Total</b>	<b>150</b>	<b>150</b>	<b>300</b>	<b>25</b>
<b>Grand Total (I to VI Semesters):4500 Marks : 150 Credits</b>					

**Dissertation Assessment (16MCA601P)**

- The project shall be carried out in the same institution or in industry/R&D labs based on software tools and technologies learnt in MCA course/internship for minimum period of 16 weeks.
- The internal examiners (Project Guide with at least 2 years experience) and the external examiners shall be appointed by the College for the Final Evaluation of the project.
- Internal and external examiners shall carry out the evaluation of Dissertation report for 75 marks individually.
- Dissertation Assessment by the internal examiner will be considered as **CIE** and Dissertation Assessment by the external examiner will be considered as **SEE**

**Internal Evaluation and Viva Voce(16MCA602P)**

- Internal assessment (CIE) shall be evaluated by both the **internal and external guide** for 75 marks individually. The average of the marks allotted by the internal and external guides shall be the final marks for the Internal Evaluation
- The project presentation and Viva-voce (SEE) shall be evaluated jointly by both the **internal and external examiners** for 75 marks.

## Semester - I

<b>Problem Solving Using C</b>			
[As per Choice Based Credit System (CBCS) scheme]			
<b>SEMESTER – I</b>			
Subject Code	16MCA101	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	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 basics of C programming language</p> <p>CO2: Acquire knowledge of, Various types of control statements and structures</p> <p>CO3: Analyze the performance of, arrays, functions, pointers, structures, unions, files and preprocessor directives</p> <p>CO4: Implement all the applications of C in a high-level language</p> <p>CO5: Design and apply appropriate C language constraints for solving computing problems.</p>			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module 1</b>			<b>10 Hours</b>
Algorithms, Flow Charts, C structure, Variables, Data types, Constants, Declarations, Operators, Precedence, Associativity, Order of evaluation, Type conversion, Storage classes, Programming Examples, Input and output statements – scanf, getchar, gets, printf, putchar, puts			
<b>Module 2</b>			<b>10 Hours</b>
Control Statements – if, else-if, switch, Control Structures – while, for, do-while, break and continue, goto, Programming Examples			
<b>Module 3</b>			<b>10 Hours</b>
Arrays – Single dimension, Two dimensional, Multi dimensional Arrays, Strings, Programming Examples, Functions, Categories of functions			
<b>Module 4</b>			<b>10 Hours</b>
Pointers, Pointer arithmetic, Call by value, Pointer Expression, Pointer as function arguments, recursion, Passing arrays to functions, passing strings to functions, Call by reference, Functions returning pointers, Pointers to functions, Programming, Examples, Structures and Unions – defining, declaring, initialization, accessing, comparing, operations on individual members; array of structures, structures within structures, structures and functions, pointers and structures, bit fields, Programming Examples			
<b>Module 5</b>			<b>10 Hours</b>
Files – defining, opening, closing, input and output operations, error handling, random access; Command line arguments; Dynamic Memory Allocation –definition, malloc, calloc, realloc, free, dynamic arrays; Preprocessor – definition, macro substitution, file inclusion, compiler control directives, Programming Examples			
<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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have 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>			

**Text Books**

1. Let us C, Yashwant Kanetkar, BPB Publications
2. Programming with C, Balaguruswamy

The C Programming Language, Brian W Kernighan, Dennis M Richie, PHI, 2nd Edition

**Reference Books:**

1. Programming with C, Byron Gottfried, Tata McGraw-Hill edition
2. Simplifying C, Harshal Arolkar, Sonal Jain, Wiley Publications
3. Head First C, David Griffiths, & Dawn Griffiths, O'Riley.

C Programming, Dr. Vishal M. Lichade, Dreamtech press

<b>UNIX Programming</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – I</b>			
Subject Code	16MCA102	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	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><b>CO1:</b> Understand and experience the UNIX environment, File system and hierarchy.</p> <p><b>CO2:</b> Demonstrate commands to extract, interpret data for further processing.</p> <p><b>CO3:</b> Apply commands to perform different tasks using AWK filtering for various applications</p> <p><b>CO4:</b> Analyze the usage of different shell commands, variables and analyzing the working of Source Code Control System Tool-GIT with Example</p> <p><b>CO5:</b> Evaluate different commands with sample shell scripts</p>			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module 1</b>			<b>10 Hours</b>
<p><b>Introduction of UNIX and Shell:</b> 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 &amp;&amp; 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.</p>			
<b>Module 2</b>			<b>10 Hours</b>
<p><b>UNIX File System:</b> 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.</p>			
<b>Module 3</b>			<b>10 Hours</b>
<p><b>Filters and Awk</b> Pr, head, tail, cut, paste, sort, uniq, tr commands, Filters using Regular Expression : grep &amp; sed grep, Regular Expression, egrep, fgrep, sed instruction, Line Addressing, Inserting and Changing Text, Context addressing, writing selected lines to a file, the -f option, Substitution, Properties of Regular Expressions Context addressing, writing selected lines to a file, the -f option, Substitution, Properties of Regular Expressions. Awk-Advanced Filters: Simple awk Filtering, Splitting a Line into Fields, printf, the Logical and Relational Operators, Number Processing, Variables, The -f option, BEGIN and END positional Parameters, get line, Built-in variables, Arrays, Functions, Interface with the Shell, Control Flow.</p>			
<b>Module 4</b>			<b>10 Hours</b>
<p><b>Advanced Shell Programming</b> The sh command, export, cd, the Command, expr, Conditional Parameter Substitution, Merging Streams, Shell Functions, eval, Exec Statement and</p>			

<p>Examples</p> <p><b>Source Code Control Systems- GIT</b></p> <p>Version Control Systems, Distributed version Control Systems, Advantages of GIT, DVCS Terminologies, Life Cycle, Create Operation, Create New User, Create a Bare Repository</p>	
<b>Module 5</b>	<b>10 Hours</b>
<p><b>Process and System Administration</b></p> <p>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 &amp; 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</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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have 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> <p>1. Your UNIX-The Ultimate Guide, Sumitabha Das, Tata Mc GrawHill,</p>	
<p><b>Reference Books:</b></p> <p>1. "Unix Shell Programming", Yashwant Kanetkar,  2. "Beginning Shell Scripting", Eric Foster-Johnson, John C Welch, Micah Anderson, Wrox publication.  3. UNIX: Concepts and Applications, Sumitabha Das, Tata Mc GrawHill,</p>	

<b>Web Technologies</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – I</b>			
Subject Code	16MCA103	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	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><b>CO1:</b> Develop Web applications using XHTML and various scripting languages.</p> <p><b>CO 2:</b> Build dynamic documents using Document Object Model (DOM).</p> <p><b>CO 3:</b> Design documents using markup languages and stylesheets.</p> <p><b>CO 4:</b> Justify the need for CGI programming between PERL and various mark-up Languages</p> <p><b>CO 5:</b> Build the ability to select the essential technology needed to develop and implement web Applications.</p>			
Modules			Teaching Hours
<b>Module 1</b>			<b>10 Hours</b>
<p><b>Introduction to XHTML;</b> Basic syntax, Standard structure, Basic text markup, Images, Hypertext Links. Lists, Tables, Forms, Frames.</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> <p><b>The Basics of JavaScript:</b> Overview of JavaScript, Object orientation and JavaScript, Syntactic characteristics, Primitives, operations, and expressions, Screen output and keyboard input, Control statements, Object creation and modification, Arrays, Functions, Constructors, Pattern matching using regular expressions, Errors in scripts,</p>			
<b>Module 2</b>			<b>10 Hours</b>
<p><b>Java Script 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 Event Model, The navigator Object, Dom Tree Traversal and Modification.</p> <p><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</p>			
<b>Module 3</b>			<b>10 Hours</b>
<p><b>Introduction to XML</b> Introduction, Syntax, Document Structure, Document type definitions, Namespaces, XML schemas, displaying raw XML documents, Displaying XML documents with CSS, XSLT style sheets, XML processors, Web services.</p>			
<b>Module 4</b>			<b>10 Hours</b>
<p><b>Perl and CGI Programming</b> Origins and uses of Perl, Scalars and their operations, Assignment statements and simple input and output, Control statements, Fundamentals of arrays,</p>			



<p>Hashes, References, Functions, Pattern matching, File input and output; Examples.</p> <p><b>Using Perl for CGI Programming:</b> The Common Gateway Interface; CGI linkage; Query string format; CGI.pm module.</p> <p><b>Building Web Applications with Perl</b> Uploading files, Using Relational Databases.</p>	
<b>Module 5</b>	<b>10 Hours</b>
<p><b>4. Introduction to PHP</b> 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</p> <p><b>Building Web applications with PHP</b> Tracking users, cookies, sessions, Using Databases, Handling XML.</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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have 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> <p>1. Robert W. Sebesta: Programming the World Wide Web, 4th Edition, Pearson education, 2012. Chapters 2, 3, 4, 5, 6, 7, 8, 9, 11&amp;13</p>	

<b>Computer Organization</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – I</b>			
Subject Code	16MCA104	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
<b>CO1:</b> Understand the Basics of Digital System.			
<b>CO2:</b> Understand the Basics of Computer System Organization.			
<b>CO3:</b> Understand the concepts of the number system in Designing Digital System.			
<b>CO4:</b> Gain knowledge on combinational circuits and sequential circuits.			
<b>CO5:</b> Analyse the need of Logic circuits in digital system.			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module 1</b>			<b>10 Hours</b>
<b>Binary Systems and Combinational Logic</b> Digital Computers and Digital Systems, Binary Numbers, Number Base Conversion, Octal and Hexadecimal Numbers, subtraction using r's and (r-1)'s complements, Binary Code, Binary Storage and Registers, Binary Logic. Axiomatic Definition of Boolean Algebra, Basic Theorems and Properties of Boolean Algebra, Boolean Functions, Canonical and Standard Forms, Other Logic Operations, The map Method, Two–and Three–Variable Maps, Four–Variables Map, Don't Care Conditions.			
<b>Module 2</b>			<b>10 Hours</b>
<b>Arithmetic Circuits</b> Digital Logic Gates, NAND and NOR Implementation, Introduction, Adders, Subtractors, Binary Parallel Adder, Decimal Adder, Magnitude Comparator, Decoders, Multiplexers, BOOTH algorithm for signed numbers with example.			
<b>Module 3</b>			<b>10 Hours</b>
<b>Sequential Logic</b> Introduction, different types of Flip– Flops, Triggering of Flip- Flops, Registers, Shift Registers, Ripple counter and Synchronous Counter. <b>Machine Instruction:</b> Introduction to Assemblers and Compilers, Memory Locations and Addresses, Memory Operations, Instructions and Instruction Sequencing, Addressing Modes.			
<b>Module 4</b>			<b>10 Hours</b>
<b>Assembly Language and Input/ Output Organization</b> Basics of Assembly Language Programme, Examples from Assembly Language Programming. Accessing I/O Devices, Interrupts, DMA, Processor Example, Buses.			
<b>Module 5</b>			<b>10 Hours</b>
<b>The Memory System</b> Basic Concepts, Semiconductor RAM Memories, Read– Only Memories, Speed, Size, and Cost, Cache Memories, Virtual Memories, Memory Management Requirements, Secondary Storage.			
<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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have questions covering all the topics under a module.</li> </ul>			
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. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, "Computer Organization", 5<sup>th</sup> edition, Tata Mc Graw-Hill, 2011

**Reference Books:**

1. John P. Hayes, "Computer Architecture and Organization", Tata Mc Graw- Hill, Edition, 2012.
2. Soumitrs Kumar Mandal, "Digital Electronics Principles and Applications", Tata Mc Graw-Hill, 2010
3. Hamacher, " Computer Organization" , McGraw-Hill Education

<b>Professional Communication and management</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – I</b>			
Subject Code	16MCA105	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b> CO1: Understanding the process of communication and its importance. CO2: Understanding how to write effective reports, Resumes, Letters, etc. CO3: Apply effective presentation strategies and group communication. CO4: Analyze motivation and leadership qualities. CO5: Understand the importance of ethics as an IT professional.			
Modules			Teaching Hours
<b>Module 1</b>			<b>10 Hours</b>
<b>Communication in the workplace-</b> Role of communication in Business, Process of Human Communication. <b>Informal Communication-</b> Listening, Non verbal Communication. <b>Correctness Of Communication-</b> Nature of correctness, Standard for Punctuation, Standards for Grammar, Standards for the use of numbers, Spelling, Capitalization.			
<b>Module 2</b>			<b>10 Hours</b>
<b>Writing for Effect-</b> Business Etiquette, conversational style, view point, positive language, and courtesy. <b>Basics Report Writing-</b> Defining Reports, Determine the Purpose, Determine the Factors, Gathering The Information, Organizing the Report, Writing The Report. <b>Physical Presentation Of Letters, Memos, And Reports-</b> Basics for all document preparation, Form of Business Letters. <b>Strategies in the job search Process-</b> Preparing application, Resume, Cover letter, Facing an Interview.			
<b>Module 3</b>			<b>10 Hours</b>
<b>Group Communication-</b> Introduction, Group discussion, Organizational group discussion, group discussion as part of selection process meetings, conferences. <b>Effective Presentation strategies-</b> Introduction, Defining purpose, Analyzing audience and locale, organizing contents, preparing outline, Visual Aids, Understanding Nuances of delivery, Kinesics, Proxemics, Paralinguistics, Chronemics, Sample speech.			
<b>Module 4</b>			<b>10 Hours</b>
<b>Motivation-</b> Motivation and Motivators, Motivation: The Carrot and the stick, The Hierarchy of needs Theory, The Motivation – Hygiene Approach to Motivation. <b>Leadership-</b> Defining Leadership, Ingredients of Leadership, Triat Approaches to Leadership, Leadership behavior and styles, Situational or contingency, Approaches to Leadership.			
<b>Module 5</b>			<b>10 Hours</b>
<b>Ethics: An overview of Ethics-</b> What are Ethics? Ethics in the business World, Ethics in Information Technology (IT). <b>Ethics for IT Professionals and IT users-</b> IT professionals, The Ethical behavior of IT professionals, IT Users.			
<b>Question paper pattern:</b>			

- The question paper will have ten questions.
- Each full question consists of 20 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. Meenakshi Raman and Sangeeta Sharma: Technical Communication-Principles and Practices, Oxford University Press, 2004.
2. George Reynolds: Ethics in Information Technology, 2<sup>nd</sup> Edition, Thomson Course Technology, 2007
3. Lesikar and Flatley: Communication-Basic Business Communication Skills for Empowering the Internet Generation 9<sup>th</sup> Edition, Tata McGraw-Hill Edition.
4. Harold Koonlz and Heinz Weihrich: Management-Essentials of Management, 5<sup>th</sup> Edition, McGraw-Hill International Edition.

<b>C Laboratory</b>			
[As per Choice Based Credit System (CBCS) scheme]			
<b>SEMESTER – I</b>			
Subject Code	16MCA106L	CIE Marks	50
Number of Lecture Hours/Week	03 Hrs Laboratory	SEE Marks	50
		SEE Hours	03
CREDITS – 1.5			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
<b>CO1:</b> Apply C concepts to develop interactive applications in C.			
<b>CO2:</b> Be fluent in the use of input output statements, constants, variables ,expressions and functions			
<b>CO3:</b> Be fluent in the use control structures, arrays ,structures and pointers			
<b>CO4:</b> Be fluent in the use of file handling techniques.			
<b>Laboratory Experiments:</b>			
<b>SECTION A</b>			
1. a. Check whether the given number is perfect number <b>Defn:</b> A perfect number is a positive integer that is equal to the sum of its proper positive divisors, that is, the sum of its positive divisors excluding the number itself. <b>Example</b> - The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and $1 + 2 + 3 = 6$ . b. Solve quadratic equations for the given values of a,b,c.			
2. Write a menu driven C program to a. Insert an element into an array b. Delete an element from the array (first occurrence)			
3. Write a C program to find the saddle point of a matrix. <b>Defn:</b> Given a RxC Matrix, A, i.e. R rows and C columns we define a Saddle-Point as Saddle_Pt (A) for a row I and column j is that A(i,j) that is the minimum of Row i and the maximum of Col j.			
4. Write a Menu driven C program to a. Accept two numbers n and m b. Sum of all integers ranging from n to m c. Sum of all odd integers ranging from n to m d. Sum of all even integers ranging from n to m Display an error message if $n > m$ . Create functions for each of the options			
5. Write a Menu Driven C Program to implement the following using recursion a. Factorial of a number b. Fibonacci series			
6. Create a structure Complex Number having real and imaginary part as properties. Write functions to add and subtract the two complex numbers.			
7. Write a menu driven C Program a .to copy two strings b. to compare two strings c. to reverse a string using pointers and not using any library functions. b. Write a C Program to compare two strings without using library function. Make use of pointers.			
8. a. Write a C Program to remove all white spaces and newline characters from a file. b. Find whether a given word exists in the file. If it exists display the location of the word			
<b>SECTION B</b>			
<b>Develop a project using the C language and concepts learnt in the theory and exercises listed in part A with a good look and feel effects.</b>			
Note:			
1. In the examination each student picks one question from part A.			

2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report that must include the following
  - a. Source Code
  - b. Sample output
4. The report must be evaluated for 5 Marks. Demonstration and Viva for 15 Marks.

**Instructions:**

1. In the examination, one exercise from part A is to be asked for 30 marks.
2. Mini project student group size is limited to Three students only.
3. The mini project under part B has to be evaluated for 20 marks.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.

**Sample suggestion for the Project:**

Student Database project: Define a structure called student having the properties of student\_id, student name and branch of the student with a sub structure of marks of 3 subjects. Write a Menu Driven C Program to

- a. Add new student detail
- b. Delete a student detail
- c. Display all student details
- d. Display the name of the student with the best mark
- e. Display the name of the student with the worst mark
- f. Display the average marks scored by the students

*Note: In the examination each student should do one question out of the above 8 questions*

<b>UNIX Programming Laboratory</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – I</b>			
Subject Code	16MCA107L	CIE Marks	50
Number of Lecture Hours/Week	03 Hrs Laboratory	SEE Marks	50
		SEE Hours	03
<b>CREDITS – 1.5</b>			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
<b>CO1:</b> Understand the Unix programming environment.			
<b>CO2:</b> Be fluent in the use of Vi editor.			
<b>CO3:</b> Be able to design and implement shell scripts to manage users with different types of permission and file based applications.			
<b>CO4:</b> Be fluent to write Awk scripts.			
<b>Laboratory Experiments:</b>		<b>PART - A</b>	
<b>A. Explore the Unix environment and</b>			
<b>Explore vi editor with vim tutor. Perform the following operations using vi editor, but not limited to:</b>			
<b>1. Insert character, delete character, replace character</b>			
<b>2. save the file and continue working</b>			
<b>3. save the file and exit the editor</b>			
<b>4. quit the editor</b>			
<b>5. quit without saving the file</b>			
<b>6. rename a file</b>			
<b>7. insert lines, delete lines,</b>			
<b>8. setline numbers</b>			
<b>9. search for a pattern</b>			
<b>10. move forward and backward</b>			
<b>1. Write a shell script that takes a valid directory name as an argument and recursively descend all the subdirectories, finds the maximum length of any file in that hierarchy and writes this maximum value to the standard output.</b>			
<b>2. Write shell script to implement terminal locking (similar to the lock command). It should prompt the user for a password. After accepting the password entered by the user, it must prompt again for the matching password as confirmation and if match occurs, it must lock the keyword until a matching password is entered again by the user, Note that the script must be written to disregard BREAK, control-D. No time limit need be implemented for the lock duration.</b>			
<b>3. Write a shell script that displays all the links to a file specified as the first argument to the script. The second argument, which is optional, can be used to specify in which the search is to begin. If this second argument is not present, the search is to begin in current working directory. In either case, the starting directory as well as all its sub directories at all levels must be searched. The script need not include any error checking.</b>			
<b>4. Write a shell script to display the calendar for current month with current date replaced by * or ** depending on whether the date has one digit or two digits</b>			
<b>5. Write a shell script to list all the files in a directory whose filename is at least 10 characters. (us expr command to check the length)</b>			
<b>6. Write a shell script that accept a list of filenames as its argument, count and report occurrence of each word that is present in the first argument file on other Argument files.</b>			
<b>7. Write a shell script that reports the logging in of a specified user within one minute after he/she login. The script automatically terminate if specified user does not login during a specified period</b>			



of time.

**8.** Write a shell script that folds long lines into 40 columns. Thus any line that exceeds 40 characters must be broken after 40th, a “\” is to be appended as the indication of folding and the processing is to be continued with the residue. The input is to be supplied through a text file created by the user.

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

**10.** 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.

### ***PART- B***

**Develop a project using the Unix concepts learnt in the theory and exercises listed in part A with a good look and feel effects.**

Note:

1. In the examination each student picks one question from part A.
2. A team of two or three students must develop the mini project. However during the examination, each student must demonstrate the project individually.
3. The team must submit a brief project report that must include the following
  - a. Source Code
  - b. Sample output
4. The report must be evaluated for 5 Marks. Demonstration and Viva for 15 Marks.

#### **Instructions:**

1. In the examination, one exercise from part A is to be asked for 30 marks.
2. Mini project student group size is limited to Three students only.
3. The mini project under part B has to be evaluated for 20 marks.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.

#### **Sample suggestion for the Project:**

Developing Shells

Developing Editors with all basic properties

***Note: In the examination each student should choose one out of the 10 questions from PART-A***

<b>Web Designing Laboratory</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – I</b>			
Subject Code	16MCA108L	CIE Marks	50
Number of Lecture Hours/Week	03 Hrs Laboratory	SEE Marks	50
		SEE Hours	03
<b>CREDITS – 1.5</b>			
<p><b>Course Outcome (CO): At the end of this course, the students will be able to</b></p> <p><b>CO1:</b> Understand the concept and usages web based programming techniques.</p> <p><b>CO2:</b> Be fluent in the use of XHTML programs using Javascript.</p> <p><b>CO3:</b> Be fluent in the use of CGI and Perl programs for different types of server side applications.</p> <p><b>CO4:</b> Be fluent in developing Web Applications using PHP.</p> <p><b>CO5:</b> Design and implement user interactive dynamic web based applications.</p>			
<b><i>PART –A (Laboratory Experiments)</i></b>			
<p>1a) Develop and demonstrate a XHTML file that includes JavaScript script for the following problems:</p> <p>i) Input: A number n obtained using prompt Output: The first n Fibonacci numbers</p> <p>ii) Input: A number n obtained using prompt Output: A table of numbers from 1 to n and their squares using <b>alert</b></p> <p>1b) Develop and demonstrate using JavaScript, a XHTML document that displays random numbers (integers).</p>			
<p>2a) Develop and demonstrate, using JavaScript script, a XHTML document that collects the USN (the valid format is: A digit from 1 to 4 followed by two uppercase characters followed by two digits followed by two uppercase characters followed by three digits; No embedded spaces allowed) of the user. Event handler must be included for the Form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.</p> <p>b) Modify the above program to get the current semester also (restricted to be a number from 1 to 6).</p>			
<p>3a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, 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.</p> <p>b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.</p>			
<p>4a) Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.</p> <p>4b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.</p>			
<p>5a) Write a PERL program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.</p> <p>5b) Create XHTML form with Name, address line1, address line2 and email text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on name.</p>			
<p>5. Write a PHP program to read student data from an XML file and store into the MySQL database. Retrieve and display using SEARCH function.</p>			

### **Part-B**

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

Note:

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

**Instructions:**

1. In the examination, one exercise from part A is to be asked for 30 marks.
2. Mini project student group size is limited to 2 or 3 students only.
3. The mini project under part B has to be evaluated for 20 marks.
4. Project report duly signed by the Guide and HOD need to be submitted during examination.

*Note: In the examination each student should choose one out of the 6 questions from PART-A*

## Semester - II

<b>Object Oriented Programming Using C++</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – II</b>			
Subject Code	16MCA201	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b> CO1: Apply C++ features to program design and implementation. CO2: Use C++ to demonstrate practical experience in developing object-oriented solutions. CO3: Analyze a problem description and build object-oriented software using good coding practices and techniques. CO4: Implement an achievable practical application and analyze issues related to object-oriented techniques in the C++ programming language.			
Modules			Teaching Hours
<b>Module 1</b>			<b>10 Hours</b>
<b>Introduction to OOPs, Modular Programming with Functions</b> Object Oriented paradigm, Structured vs. Object Oriented Paradigm. Elements of Object Oriented Programming: Object, Classes, Encapsulation & data abstraction, Inheritance, Polymorphism etc., C++ Overview, different data types, operators, expressions, const& Volatile qualifiers, arrays and strings, reference variables. <b>Modular Programming with Functions:</b> Function Components, argument passing, inline functions, function templates, recursive functions			
<b>Module 2</b>			<b>10 Hours</b>
<b>Classes &amp; Objects and Operator Overloading</b> Introduction, Class Specification, Class Objects access members, defining member functions, Data hiding, constructors, destructors, parameterized constructors, static data members, Functions, scope resolution operator, passing objects as arguments, returning objects, friend Functions & classes, arrays of objects, Dynamic objects – Pointers to objects, Class members. Creating a Member Operator function, Binary operator overloading, Concatenation of strings, strings comparison using operator overloading, overloading the assignment operator ,overloading operators such as [], ->, increment & decrement operators, Operator overloading using friend functions +,-, overloading input stream and output stream.			
<b>Module 3</b>			<b>10 Hours</b>
<b>Inheritance:</b> Basic Concepts, Reusability & Extensibility. Defining derived classes, protected access specifier in Base class – public, private & protected inheritance – constructors and Destructors in derived classes – Types of Inheritances. Virtual base class. <b>Virtual functions:</b> Normal member functions accessed with pointers, virtual member function access, late binding, pure virtual function, abstract classes. <b>Polymorphism:</b> Overloading Concepts Function Overloading: Functions with different sets of parameter default and constant parameters. <b>Operator Overloading:</b> Defining Operator Function, Rules for overloading Operators. Overloading unary operators, overloading binary Operators, Overloading Comma, [], (), ->, new, delete Operators. <b>Type Conversions:</b> Basic to class, Class to Basic and one Class to another Class type. Advanced Typecasting.			

<b>Module 4</b>	<b>10 Hours</b>
<p><b>Templates, Exception Handling and I/O Streams</b> Generic classes, a class template with more than one generic type, the power of templates.</p> <p><b>Namespaces and Conversion Functions:</b> Namespace fundamentals, using keyword, unnamed namespaces, const member function and mutable, Volatile member functions</p> <p><b>Exception Handling:</b> Exception handling model, Exception handling constructs, list of Exceptions catch all exceptions, handling uncaught exceptions.</p>	
<b>Module 5</b>	<b>10 Hours</b>
<p><b>I/O Streams:</b> IO Stream basics, Output operator &lt;&lt;, input &gt;&gt;, additional I/O operators, overloading the output operator &lt;&lt;, Overloading the input operator &gt;&gt;, file input &amp; output, manipulators</p> <p><b>Run Time Type ID and the Casting Operators:</b> Run Time Type Identification(RTTI), the Casting Operator, dynamic_cast</p> <p><b>STL</b> STL: An overview, containers, vectors, lists, maps</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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have 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. K.R. Venugopal, RajkumarBuyya, T. Ravishankar: Mastering C++ ,TataMcGraw Hill Publication, 2006.</li> <li>2. Herbert Schildt: C++ The Complete Reference, 4th Edition, Tata McGraw Hill, 2007.</li> <li>3. Object Oriented Programming in C++:By M.T Somashekara ,D.S Guru,H.S. Nagendraswamy, K.SManjunatha2nd Editions 2012</li> </ol>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Stephen Prata : C++ Primer Plus, 6th Edition, Pearson Education.</li> <li>2. Al Stevens: C++ Programming, 7th Edition, Wiley India Publications</li> <li>3. Stanley B.Lippmann, JoseeLajore: C++Primer, 4th Edition, Addison Wesley, 2005.</li> <li>4. Object oriented programming with C++, E. Balaguruswamy, TMH.</li> </ol>	

<b>Data Structures using C</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – II</b>			
Subject Code	16MCA202	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
<b>Credits – 04</b>			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
<b>CO1:</b> Recall the concepts of Pointers, structures, strings, arrays and their representations			
<b>CO2:</b> Acquire knowledge of			
- Various types of data structures as ADT.			
- Sorting and searching operations			
<b>CO3:</b> Analyze the performance of			
- Stack, Queue, Lists, Trees, Hashing, Searching and Sorting techniques			
<b>CO4:</b> Implement all the applications of Data structures in a high-level language			
<b>CO5:</b> Design and apply appropriate data structures for solving computing problems.			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module 1</b>			<b>10 Hours</b>
<b>Introduction to Data Structures</b>			
<b>Classification of Data Structures:</b> Primitive and Non- Primitive, Linear and Nonlinear; Data structure Operations <b>Pointers:</b> Pointer Expression, Pointer as function arguments, Functions returning pointers, Pointers to functions. <b>Structures:</b> Declaring and using structure types. <b>ADT:</b> Array as ADT, Operations - Insert, Delete, Search, Sort. <b>String:</b> Definition, Representation, String as ADT, Operations – Insert, Delete, Concatenate, Comparing, Substring. <b>Recursion</b> - Factorial, GCD, Fibonacci Sequence. <b>Algorithm Complexity, Space Complexity &amp; Time Complexity.</b>			
<b>Module 2</b>			<b>10 Hours</b>
<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>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>10 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 lists.			
<b>Module 4</b>			<b>10 Hours</b>
<b>Double Linked List:</b> Inserting and Deleting Nodes, Queue as doubly linked lists, such as insert into position, Delete a specified element. <b>Application of Linked Lists:</b> Stacks, Queues, Double-ended Queues, Priority Queues, Trees, BST. <b>Application of DLL to memory</b>			

<b>management. Graph Data Structure:</b> Basics, Operations- Add/Remove Vertex, Add/Remove Edge, Display	
<b>Module 5</b>	<b>10 Hours</b>
<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. AVL Tree and Operations, Red-Black Tree.	
<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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have 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>  <b>Data Structures Using C and C++</b> by Yedidyah Langsam and Moshe J. Augenstein and Aaron M Tenanbanum, 2nd Edition, Pearson Education Asia, 2002.</p>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Balaguruswamy: Data Structures Using C, McGraw Hill Education</li> <li>2. Data Structures and Algorithm Analysis in C, Mark Allen Weiss, 2<sup>nd</sup> Edition, Pearson Education Aisa, 1997.</li> <li>3. Richard F Giberg and Behrouz A Forouzan: Data Structures – A Pseudo code Approach with C, 2nd Edition, Cengage Learning</li> </ol> <p>Robert Kruse, C L Tondo, Bruce Leung and Shashi Mogalla: Data Structures and Program Design in C, 2nd Edition, Pearson Education</p>	

<b>DATABASE MANAGEMENT SYSTEM</b>			
[As per Choice Based Credit System (CBCS) scheme]			
<b>SEMESTER – II</b>			
Subject Code	16MCA203	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
<b>Credits – 04</b>			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
<b>CO1:</b> Demonstrate the fundamentals of data models and conceptualize and depict a database system and Make use of ER diagram in developing ER Model			
<b>CO2:</b> To Summarize the SQL and relational database design.			
<b>CO3:</b> Illustrate transaction processing, concurrency control techniques and recovery			
<b>CO4:</b> Inference the database design in the real world entities.			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module 1</b>			<b>10 Hours</b>
<b>Introduction:</b> 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.			
<b>Module 2</b>			<b>10 Hours</b>
<b>Relational Model</b> Relational Model Concepts and Constraints, 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			
<b>Module 3</b>			<b>10 Hours</b>
<b>Introduction to SQL:</b> 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.			
<b>Module 4</b>			<b>10 Hours</b>
<b>Database Design:</b> 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, Multivalued Dependencies and IV Normal Forms, Join Dependencies and V Normal Forms, Inference Rules, Equivalence and Minimal			



Cover.	
<b>Module 5</b>	<b>10 Hours</b>
<p><b>Transaction Management:</b>  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.</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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have 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. Elmasri and Navathe: Fundamentals of Database Systems, 5th Edition, Addison -Wesley, 2011. Silberschatz, Korth and Sudharshan Data base System Concepts,6<sup>th</sup> Edition, Tata McGraw Hill, 2011.</li> </ol>	
<p><b>Reference Books:</b></p> <ol style="list-style-type: none"> <li>1. Raghu Ramakrishnan and Johannes Gehrke: Database Management Systems, 3<sup>rd</sup> Edition, McGraw-Hill, 2003</li> <li>2. Silberschatz, Korth and Sudharshan: Data base System Concepts, 5th Edition, McGrawHill, 2006.</li> <li>3. Database Principles Fundamentals of Design, Implementation and Management by Coronel, Morris, Rob- Cengage Learning 2012</li> </ol>	

<b>Operating Systems</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – II</b>			
Subject Code	16MCA204	CIE Marks	50
Number of Lecture Hours/Week	04	SEE Marks	50
Total Number of Lecture Hours	50	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 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.			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module 1</b>			<b>10 Hours</b>
<b>Operating Systems Structure</b> Operating System operations, Registers, Memory Hierarchy, Cache Memory, Introduction to Operating System, Mainframe Systems, Desktop Systems, Multiprocessor Systems, Distributed Systems, Clustered Systems, Real- Time Systems, Handheld Systems, System Components, System Calls, System Programs, System Structure, Virtual Machines			
<b>Module 2</b>			<b>10 Hours</b>
<b>Process Management and Threads</b> Process States, Process Management, IPC, Threads, Overview of Threads, Symmetric Multiprocessing (SMP), Scheduling criteria; Scheduling algorithms. Principles of Concurrency, Mutual Exclusion, Hardware Support: Semaphores, Monitors, Readers/Writes Problem.			
<b>Module 3</b>			<b>10 Hours</b>
<b>Deadlock and Memory Management:</b> Principles of Deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Dining Philosophers Problem, Memory Management: Swapping, Contiguous Memory Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Process Creation, Page Replacement Alg,			
<b>Module 4</b>			<b>10 Hours</b>
<b>File System and Secondary Storage</b> File Concept, Access Methods, Directory Structure, File System Mounting, File Sharing, Protection, File–System Structure, File– System Implementation, Directory Implementation, Allocation Methods, Free– Space Management, Disk Structure, Disk Scheduling, Disk Management.			
<b>Module 5</b>			<b>10 Hours</b>
<b>Computer Security and Case study of Linux Operating system</b> The Security Problem, User Authentication, Program Threats, System Threats. Linux System. Linux history, Design Principles, Kernel modules, Process management, scheduling, Memory management, File systems, Input and output, Inter-process communications			
<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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have questions covering all the topics under a module.</li> </ul>			
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, 6<sup>th</sup> edition, 2012

**Reference Books:**

1. Chakraborty, "Operating Systems" Jaico Publishing House, 2011.
2. Dhananjay M. Dhamdhere, "Operating Systems–A Concept– Based Approach", Tata McGraw–Hill, 3<sup>rd</sup> Edition, 2012
3. Elmasri, Carrick, Levine, "Operating Systems–A Spiral Approach", Tata McGraw–Hill, 2012
4. Dhamdhere , "Operating System : A Concept", McGraw–Hill Education

<b>System Software</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – II</b>			
Subject Code	16MCA205	CIE Marks	50
Number of Lecture Hours/Week	03	SEE Marks	50
Total Number of Lecture Hours	50	SEE Hours	03
CREDITS – 04			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
<b>CO1:</b> Understand the introductory concepts of system software, SIC and SIC/XE machine architecture.			
<b>CO2:</b> Understand the design and implementation of Assemblers with implementation examples.			
<b>CO3:</b> Design and implement the linkers and loaders, macro processors and respective implementation examples.			
<b>CO4:</b> Learn the basic design and working of compilers.			
<b>Modules</b>			<b>Teaching Hours</b>
<b>Module 1</b>			<b>10 Hours</b>
<b>Machine Architecture:</b> Introduction, System Software and Machine Architecture, Simplified Instructional Computer (SIC) - SIC Machine Architecture, SIC/XE Machine Architecture, SIC Programming Examples. <b>Editors and Debugging Systems:</b> Text Editors - Overview of Editing Process, User Interface, Editor Structure, Interactive Debugging Systems - Debugging Functions and Capabilities, Relationship With Other Parts Of The System, User-Interface Criteria.			
<b>Module 2</b>			<b>10 Hours</b>
<b>Assemblers:</b> Basic Assembler Function - A Simple SIC Assembler, Assembler Algorithm and Data Structures, 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.			
<b>Module 3</b>			<b>10 Hours</b>
<b>Loaders and Linkers:</b> 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.			
<b>Module 4</b>			<b>10 Hours</b>
<b>Macro Processor:</b> 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.			
<b>Module 5</b>			<b>10 Hours</b>
<b>Compilers:</b> 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 System Software: introduction into passes, Interpreters, P-code Compilers, Compiler-Compilers.	
<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 20 marks.</li> <li>· There will be 2 questions from each module.</li> <li>· Each question will have 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> <p>1. System Software: An introduction to System Programming, Leland L. Beck &amp; D. Manjula, 3<sup>rd</sup> edition, Pearson Education, 1997.</p>	

<b>Object Oriented Programming with C ++ Laboratory</b> [As per Choice Based Credit System (CBCS) scheme] <b>SEMESTER – II</b>			
Subject Code	16MCA206L	CIE Marks	50
Number of Lecture Hours/Week	03 Hrs Laboratory	SEE Marks	50
		SEE Hours	03
<b>CREDITS – 1.5</b>			
<b>Course Outcome (CO): At the end of this course, the students will be able to</b>			
<b>CO1:</b> Apply and implement major object oriented concepts like function overloading, operator overloading, Encapsulations, and inheritance, message passing to solve real-world problems.			
<b>CO2:</b> Use major C++ features such as Virtual functions, Templates for data type independent designs and File I/O to deal with large data sets.			
<b>CO3:</b> Analyze, design and develop solutions to real-world problems applying OOP Concepts of C++.			
<b>Laboratory Experiments:</b>			
<b>NOTE: The experiments are to be carried using discrete components only.</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	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.		
4	Write a C++ program to create a class called STACK using an array of integers. Implement the following operations by overloading the operators ‘+’ and ‘--’: (i) s1 = s1 + element; where s1 is an object of the class STACK and element is an integer to be pushed on the top of the stack (ii) s1 = --s1 ; where s1 is an object of the class STACK. ‘--’ operator pops the element. Handle the STACK empty and full conditions. Also display the contents of the stack after each operation, by overloading the << operator.		
5	Write a C++ program to create a class called DATE. Accept two valid dates in the form dd/mm/yy. Implement the following operations by overloading the operators ‘+’ and ‘-’. After every operation display the results by overloading the operator <<. (i) no_of_days = d1 – d2; where d1 and d2 are DATE objects, and no_of_days is an integer (ii) d2 = d1 + no_of_days; where d1 is a DATE object and no_of_days is an integer		
6	Write a program to demonstrate friend functions and friend classes		
7	Write a C++ program to create a class template called QUEUE with member functions to add an element and to delete an element from the queue. Using the member functions, implement a queue of integers and double. Demonstrate the operations by displaying the contents of the queue after every operation.		
8	Write a C++ program to create a class called STRING and implement the following operations. Display the results after every operation by <i>overloading the operator</i>		

	<< (i) STRING s1 = "VTU" (ii) STRING s2 = "BELGAUM" (iii) STRING s3 = s1 + s2 (Use copy constructor)
9	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
10	Write a C++ program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the classes UGSTUDENT and PGSTUDENT having fields as Semester, Fees and Stipend. Enter the data for at least 5 students. Find the semester wise average age for all UG and PG students separately.
11	Write a program to create a generic stack class and member functions to perform stack operations
12	Write a program to create a file to store some records and search for a particular record and display it.
13	Write a program to implement Exception Handling with minimum 5 exceptions Classes including two built-in exceptions
14	Write a program to demonstrate virtual and pure virtual functions

**Note: In the examination each student should do one question out of the above 14 questions**

<b>Data Structures Laboratory</b> [As per Choice Based Credit System (CBCS) scheme]			
<b>SEMESTER – II</b>			
Subject Code	16MCA207L	CIE Marks	50
Number of Lecture Hours/Week	03 Hrs Laboratory	SEE Marks	50
		SEE Hours	03
<b>CREDITS – 1.5</b>			
<b>Course Outcomes(CO):</b> This laboratory course enable students to get practical experience in design, develop, implement, analyze and evaluation/testing of <b>CO1:</b> Apply data structure concepts to develop interactive applications in C. <b>CO2:</b> Linear data structures and their applications such as Stacks, Queues and Lists <b>CO3:</b> Non-Linear Data Structures and their Applications <b>CO4:</b> Be fluent in the use of different types of sorting and searching techniques			
<b>Laboratory Experiments:</b>			
1. Write a C program to implement the following using <b>recursion</b> a. Sum of n numbers b. Generate Fibonacci sequence.			
2. Implement a Program in C for Converting an <b>Infix Expression to Postfix Expression.</b>			
3. Implement a Program in C for Evaluating an <b>Postfix Expression.</b>			
4. Implement a menu driven Program in C for the following operations on <b>Circular QUEUE</b> of Integers (Array Implementation of Queue with maximum size <b>MAX</b> ) a. Insert an Element on to Circular QUEUE b. Delete an Element from Circular QUEUE c. Demonstrate <b>Overflow</b> and <b>Underflow</b> situations on Circular QUEUE d. Display the status of Circular QUEUE e. Exit			
5. Write a C program to Simulate the working of a linked list providing the following operations a. Insert at the beginning b. Insert at the end c. Delete at the beginning d. Delete at the end e. Display			
6. Write a C Program using <b>Doubly Linked List</b> to Implement Stack operations to store Integers a. <b>Push</b> an Element on to Stack b. <b>Pop</b> an Element from Stack c. Demonstrate <b>Overflow</b> and <b>Underflow</b> situations on Stack d. Display the status of Stack			
7. Implement a menu driven Program in C for the following operations on <b>Binary Tree</b> of Integers a. Create a BST of <b>N</b> Integers b. Traverse the BST in Inorder, Preorder and Post Order c. Delete an element from BST d. Exit			
8. Write a C program to Implement the following <b>searching techniques</b> a. Linear Search b. Binary Search			



<p>9. Write a C program to implement the following <b>sorting</b> algorithms using user defined functions:</p> <ol style="list-style-type: none"> <li>a. Bubble sort (Ascending order)</li> <li>b. Selection sort (Descending order)</li> </ol>
<p>10. Write a C program to sort a list of N integers using the quick sort algorithm.</p>
<p>11. Write a C program using dynamic variables and pointers, to construct a singly linked list consisting of the following information in each node: student id (integer), student name (character string) and semester (integer). The operation to be supported are:</p> <ol style="list-style-type: none"> <li>a. The insertion operation <ol style="list-style-type: none"> <li>i. At the front of a list.</li> <li>ii. At the back of the list.</li> <li>iii. At any position in the list.</li> </ol> </li> <li>b. Deleting a node based on student id. If the specified node is not present in the list an error message should be displayed. Both the options should be demonstrated.</li> <li>c. Searching a node based on student id and updates the information content. If the specified node is not present in the list an error message should be displayed. Both situations should be displayed.</li> <li>d. Displaying all the nodes in the list.</li> </ol>
<p>12. Write a C program using dynamic variables and pointers, to construct an ordered singly linked list based on the rank of the student, where each node consists of the following information: student id, student name and rank(integer).</p>
<p>13. Write a C program to sort a given array using Quick sort.</p>
<p>14. Write a C program to perform following operations on string without using string inbuilt functions.</p> <ol style="list-style-type: none"> <li>a) Copying</li> <li>b) Concatenation</li> </ol>

***Note: In the examination each student should do one question out of the above 14 questions***

**DATABASE MANAGEMENT SYSTEMS LABORATORY**

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

**SEMESTER – II**

Subject Code	16MCA208L	CIE Marks	50
Number of Lecture Hours/Week	03 Hrs Laboratory	SEE Marks	50
		SEE Hours	03

**CREDITS – 1.5****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

***Laboratory Experiments:*****1. Design an ER-diagram for the following scenario, Convert the same into a relational model and then solve the following queries.**

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

**Queries**

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

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

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

multiple projects. Each project is worked on by one or more graduate students (known as the project's research assistants). Graduate students can work on multiple projects. Each professor can supervise many students. A student who is working on a project can be supervised by only one professor.

**Queries**

1. Retrieve the names of all professors who do not have an ongoing project of more than 1Lakh.
2. Retrieve the names of all graduate students along with their professors under whom they Work and project sponsor.
3. List the professors and sum of the budget of their projects started after 2012 but ended in 2014
4. List the names of professors who has a total worth of project greater than the average budget of projects sanctioned
5. List the professors who work on all the projects.

**3. Consider the following Scenario and design an ER-Diagram by identifying Weak entity type and map the designed ER-diagram into a Relational model and solve the following queries. :**

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

**Queries:**

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

**4. Design an ER-diagram for the following scenario, by identifying a ternary relationship and Convert the same into a relational model, then solve the following queries.**

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

**Queries:**

- a) List the details of patients who are 20 years old and have been checked by eye-specialist.
- b) List the details of doctors who have given the prescription to more than 5 patients in year 2013.
- c) List the details of pharmaceutical companies who supply drug to more than 5 pharmacies in the same city where company is located.
- d) List the details of drug supplied by only one pharmaceutical company.

e) List the details of drug supplied by all pharmaceutical companies.

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

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

**Queries:**

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

**Note 1: In the practical Examination each student has to pick one question from a lot of all the 5 questions.**

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