



ST ALOYSIUS COLLEGE(AUTONOMOUS)

NATIONAL EDUCATION POLICY - 2020 (NEP-2020)

Curriculum Structures
for

Bachelor of Science

Basic and Honours Programmes

with

Computer Science as Major Course

Syllabus for I and II Semesters

and

Open Elective Courses in Computer Science



Re-accredited by NAAC with 'A' Grade with CGPA 3.62/4
Recognised by UGC as "College with Potential for Excellence"
Conferred "College with "STAR STATUS" by DBT, Government of India.
Centre for Research Capacity Building under UGC-STRIDE

Date: 17-08-2022

NOTIFICATION

Sub: Syllabus of **B.Se. COMPUTER SCIENCE** under NEP Regulations, 2021.
(As per Mangalore University guidelines)

- Ref: 1. Decision of the Academic Council meeting held on 18-12-2021 vide
Agenda No: 6.25(2021-22)
2. Decision of the Academic Council meeting held on 09-07-2022 vide
Agenda No: 14
3. Office Notification dated 21-02-2022
4. Office Notification dated 17-08-2022

Pursuant to the above, the Syllabus of **B.Se. COMPUTER SCIENCE** under NEP Regulations, 2021 which was approved by the Academic Council at its meeting held on 18-12-2021 & 09-07-2022 is hereby notified for implementation with effect from the academic year 2021-22.

PRINCIPAL



REGISTRAR

To:

1. The Chairman/Dean/HOD.
2. The Registrar Office
3. Library

Index

S. No.	Content	Page
1	Objective and program outcome	3
2	Model Programme Structure	5
3	Curriculum Structure	5 – 6
4	Course Content DSC-1	7 – 8
5	Course Content DSC-1 Lab	9 – 10
6	Course Content CSOE1 / CAO E1	11 – 12
7	Course Content DSC-2	13 – 14
8	Course Content DSC-2 Lab	15 – 16
9	Course Content CSOE2 / CAO E2	17 – 18
10	Scheme of Assessment for Theory Examination	19

The objectives of the Programme are:

1. The primary objective of this program is to provide a foundation of computing principles and business practices for effectively using/managing information systems and enterprise software
2. It helps students analyze the requirements for system development and exposes students to business software and information systems
3. This course provides students with options to specialize in legacy application software, system software or mobile applications
4. To produce outstanding IT professionals who can apply the theoretical knowledge into practice in the real world and develop standalone live projects themselves
5. To provide opportunity for the study of modern methods of information processing and its applications.
6. To develop among students the programming techniques and the problem-solving skills through programming
7. To prepare students who wish to go on to further studies in computer science and related subjects.
8. To acquaint students to Work effectively with a range of current, standard, Office Productivity software applications.

Program Outcomes:

1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity
2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyze problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.
3. **Design and Development of Solutions:** Ability to design and development of algorithmic solutions to real world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems.
4. **Programming a computer:** Exhibiting strong skills required to program a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.
5. **Application Systems Knowledge:** Possessing a sound knowledge on computer application

software and ability to design and develop app for applicative problems.

6. **Modern Tool Usage:** Identify, select and use a modern scientific and IT tool or technique for modeling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.
7. **Communication:** Must have a reasonably good communication knowledge both in oral and writing.
8. **Project Management:** Practicing of existing projects and becoming independent to launch own project by identifying a gap in solutions.
9. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrity in a working environment and also have concern on societal impacts due to computer-based solutions for problems.
10. **Lifelong Learning:** Should become an independent learner. So, learn to learn ability.
11. **Motivation to take up Higher Studies:** Inspiration to continue educations towards advanced studies on Computer Science.

Syllabus Structure of Computer Science Paper as one of the major papers and open elective papers for BSc (Computer Science).

Subject Code	sem	subject	Theory hours/week	Practical hours/week	Duration of exams	Marks and credits			
						IA	Exam	Total	Credits
G505DC1.1	I	Computer Fundamentals and Programming in C	4		03	40	60	100	4
G505DC1.1P	I	C Programming Lab		4	02	25	25	50	2
G505OE1.1	I	Office Automation	3		03	40	60	100	3
G505DC2.2	II	Data Structures using C	4		03	40	60	100	4
G505DC2.2P	II	Data Structures Lab		4	03	25	25	50	2
G505OE2.2	II	Web Designing	3		03	40	60	100	3
G505DC3.3	III	object oriented Programming concepts and programming JAVA	4		03	40	60	100	4
G505DC3.3P	III	java lab		4	02	25	25	50	2
G505OE3.3	III	Internet of things	3		03	40	60	100	3
G505DC4.4	IV	Data Base Management System	4		03	40	60	100	4
G505DC4.4P	IV	DBMS lab		4	02	25	25	50	2

Curriculum Structure

Program: B.Sc. (Basic and Honors)

Subject: Computer Science

Sem	Discipline Specific Core Courses (DSC)	Hours/Week		Discipline Specific Elective Courses (DSE)/ Vocational Courses (VC)	Hours/Week
		Theory	Lab		
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4		
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4		
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4		
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4		
5	DSC-5: Programming in PYTHON DSC-6: Operating System Concepts DSC-5Lab: PYTHON Programming lab DSC-6Lab: Operating System lab	3 3	4 4	VC-1: Any one from Vocational Courses, Group – 1*	3
6	DSC-7: Internet Technologies DSC-8: Computer Networks DSC-7Lab: JAVA Script, HTML, CSS Lab DSC-8Lab: Computer Networks Lab	3 3	4 4	VC-2: Any one from Vocational Courses, Group – 2* Internship:	3
7	DSC-9: Computer Graphics and Visualization DSC-10: Design and Analysis of Algorithms DSC-11: Software Engineering DSC-9Lab: Computer Graphics and Visualization Lab DSC-10Lab: Algorithms Lab	3 3 3	4 4	DSE-1: Any one from Discipline Specific Elective Courses, Group – 1** DSE-2: Any one from Discipline Specific Elective Courses, Group – 2** Research Methodology:	3 3 3
8	DSC-12: Artificial Intelligence and Applications DSC-13: Computer Organization and Architecture DSC-14: Data Warehousing and Data Mining	3 3 3		DSE-3: DSE-4: Any two from Discipline Specific Elective Courses, Group – 3 Research Project:	3 3 6

*** Vocational Courses**

<p style="text-align: center;">Group-1</p> <ul style="list-style-type: none">• DTP, CAD and Multimedia• Hardware and Server Maintenance• Web Content Management Systems• E-Commerce• Web Designing
<p style="text-align: center;">Group-2</p> <ul style="list-style-type: none">• Health Care Technologies• Digital Marketing• Office Automation• Multimedia Processing• Accounting Package

**** Discipline Specific Elective Courses**

<p style="text-align: center;">Group-1</p> <ul style="list-style-type: none">• IoT• Cyber Law and Cyber Security• Web Programming - PHP and MySQL• Clouds, Grids, and Clusters• Software Testing
<p style="text-align: center;">Group-2</p> <ul style="list-style-type: none">• Information and Network Security• Data Compression• Discrete Structures• Open source Programming• Multimedia Computing• Big Data
<p style="text-align: center;">Group-3</p> <ul style="list-style-type: none">• Data Analytics• Storage Area Networks• Pattern Recognition• Digital Image Processing• Parallel Programming• Digital Signal Processing

First Semester

Course Code: G505DC1.1	Course Title: Computer Fundamentals and Programming in C
Course Credits: 4	Hours of Teaching/Week: 4
Total Contact Hours: 52	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hours

Course Outcomes (COs)

After completing this course satisfactorily, a student will be able to:

- Operate desktop computers to carry out computational tasks
- Understand working of hardware and software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Course Content

Content	Hours
Unit - 1	
Fundamentals of Computers: Introduction to Computers –Definition of a computer, Characteristics of computers, Evolution of computers, Generations of computers, Classification of computers. Computer system, applications of computers. Number Systems – different types, conversion from one number system to another; Coding schemes –ASCII and Unicode. Computer Software – Categories of software. Computer Programming and Languages –Machine Level, Assembly level and Highlevel languages; Translator Programs – Assembler, Interpreter and Compiler. Developing a computer program, Program Development Cycle - Algorithm, Flowchart and Pseudocode with examples. Introduction to C Programming: Overview of C; History and Importance of C; Structure of a C Program with Examples; Creating and Executing a C Program; Compilation process in C.	13
Unit - 2	
C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; Data types; Declaration and initialization of variables; Symbolic constants. C Operators and Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity. Evaluation of arithmetic expressions; Type conversion. Input and output with C: Formatted I/O functions - <i>printf</i> and <i>scanf</i> , control stings and escape sequences, output specifications with <i>printf</i> functions; Unformatted I/O functions to read and display single character and a string - <i>getchar</i> , <i>putchar</i> , <i>gets</i> and <i>puts</i> functions.	13

Unit - 3	
<p>Control Structures: Branching: if, if-else, nested if, else-if ladder, switch. Looping: while, do-while and for loop, nested loops, exit, break, jumps in loops.</p> <p>Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation.</p> <p>Strings: Declaring and Initializing string variables; String handling functions - <i>strlen</i>, <i>strcmp</i>, <i>strcpy</i> and <i>strcat</i>; Character handling functions - <i>tolower</i>, <i>toupper</i>, <i>isalpha</i>, <i>isnumeric</i> etc.</p>	13
Unit - 4	
<p>Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers.</p> <p>User-Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type.</p> <p>User-Defined Data Types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.</p>	13

Text Books:

1. Pradeep K. Sinha and Priti Sinha: **Computer Fundamentals** (Sixth Edition), BPB Publication.
2. ITL Education Solution Limited, **Introduction to Information Technology**, Second Edition 2018, Pearson Education.
3. E. Balagurusamy: **Programming in ANSI C** (TMH), 7th Edition.

Reference Books :

1. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
2. V. Rajaraman: Programming in C (PHI – EEE)
3. S. Byron Gottfried: Programming with C (TMH)
4. Kernighan & Ritchie: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

Course Code: G505DC1.1P	Course Title: C Programming Lab
Course Credits: 2	Hours of Teaching/Week: 4
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hours

Practice Lab

The following activities be carried out in the lab during the initial period of the semester.

1. Basic Computer Proficiency
 - a) Familiarization of Computer Hardware.
 - b) Basic Computer Operations and Maintenance.
 - c) Do's and Don'ts, Safety Guidelines in Computer Lab.
2. Familiarization of Basic Software – Operating System, Word Processors, Internet Browsers, Integrated Development Environment (IDE) with Examples.
3. Type Program Code, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

Programming Lab

Part A

1. Program to read three numbers and find the biggest of three.
2. Program to find the area of a triangle using three sides of triangle.
3. Program to check for prime number.
4. Program to generate n Fibonacci numbers.
5. Program to read a multi - digit number find the sum of the digits, reverse the number and check it for palindrome.
6. Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers.
7. Program to accept student name and marks in three subjects. Find the total marks, average and grade (depending on the average marks).
8. Program to find the roots of quadratic equation(Demonstration of switch Statement).
9. Program to find largest and smallest element in a list of 'n' elements (Demonstration of one-dimensional array).
10. Program to multiply two matrices.

Part B

1. Program to accept 'N' and find the sum of the series $1! + 3! + 5! + \dots + n!$
2. Write user-defined functions to (a) find the length of a string (b) concatenate two strings. Call these functions in the main program.
3. Program to find whether a given string is palindrome or not (Use a function to reverse a string using pointers).
4. Program to transpose a matrix of order N x M and check whether it is symmetric or not.
5. Program to add two matrices using pointers.
6. Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.

7. Program to display the first 'n' Fibonacci numbers using a function to generate the nth Fibonacci number.
8. Program to find the GCD of 'n' integers using a function to compute the GCD of two integers.
9. Program to enter the information of n students (name, register number, marks in three subjects) into an array of structures. Compute and print the result of all students. For passing, student should get at least 35 in each subject, otherwise result is "FAIL".
If the student passes and if percentage ≥ 70 , result is DISTINCTION; If percentage is < 70 and ≥ 60 , result is FIRST CLASS; if percentage is < 60 and ≥ 50 , result is SECOND CLASS; otherwise result is PASS CLASS. Get the output of all students in a tabular form with proper column headings.
10. Program to prepare the pay slip of 'n' employees using an array of structures. Input the employee name, employee number and basic pay. Calculate the DA, HRA, PF, PT, Gross Pay and Net Pay as follows:
If Basic < 40000 , DA = 50% of Basic, HRA = 12% of Basic, PF = 12% of Gross Pay, PT = 250. Otherwise DA = 40% of Basic, HRA = 10% of Basic, PF = 13% of Gross, PT = 300.
Gross Pay = Basic + DA + HRA and Net Pay = Gross Pay – PF – PT.

Evaluation Scheme for Practical Examination

Assessment Criteria		Marks
Program-1 from Part A	Writing the Program	7
	Execution and Formatting	8
Program-2 from Part B	Writing the Program	8
	Execution and Formatting	12
Practical Records		10
Viva		5
Total (converted to 25)		50

Course Code: G505OE1.1	Course Title: Office Automation
Course Credits: 3	Hours of Teaching/Week: 3
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hours

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Compare and contrast various types of operating systems
- Explain the purpose of office automation
- Describe how information is stored and retrieved in/from computer memory
- Know about various types of office automation software and their applications
- Create document using word processing software
- Design presentation using presentation software
- Create worksheets using spreadsheet software
- Store and retrieve data in/from database management application

Course Content

Content	Hours
Unit - 1	
Computer software: Introduction, Software definition, Software categories, Installing and uninstalling software, Software piracy, Software terminologies Introduction to windows Operating System, operating with windows, GUI, use of help features, starting an application, essential accessories, creating shortcuts, windows explorer, control panel, finding folders and files, System utilities, memory, network basics(LAN, WAN, Man). MS-Office: Introduction, Office user interface, Microsoft office Components. MS-Word: Introduction, Starting MS-Word, Microsoft word Environment working with word documents, working with text, working with tables checking spelling and grammar, adding graphs to the document, mail merge, header and footers, page numbers, protect the document, working with formatting tools.	12
Unit - 2	
MS-Excel: Introduction, starting MS Excel, Microsoft Excel environment, Working with Excel workbook, Working with worksheet – Entering data, Excel formatting tips and Techniques, Generating graphs, Formulas and Functions, Inserting charts, Sorting, Pivot Tables, data extraction, adding clip art, add an image from a file, Printing in Excel.	10
Unit - 3	
MS-PowerPoint: Starting MS PowerPoint, Working with PowerPoint, Creating, Saving and Printing a presentation, Working with Animation, adding a slide to presentation, navigating through a presentation, Slide-sorter, Slide-show, editing slides, Working with Graphics and Multimedia in PowerPoint (Inserting Photo, Video, and Audio). The Internet: Basic internet terms, Internet applications, Internet tools, Web browser, Web browser features, Internet Explorer environment, Electronic mail, Email address structure, Advantages and disadvantages of email.	10

Unit - 4	
Database Fundamentals - Basic database terms, Database Management System. MS-Access: Introduction to Access, Creating Tables and Database, Data Type and Properties, Adding & Deleting Field in Table, Primary Key Fields, Queries, Forms: The Forms wizard saving forms, Modifying forms, Pages, Macro, Module, Reports, Printing Report, Forms.	10

Text Book:

1. ITL Education Solution Limited, Introduction to Information Technology, Second Edition., Pearson

Reference Books:

1. Peter Norton, Introduction to Computers, 7th edition, Tata McGraw Hill Publication, 2011)
2. Anita Goel, Computer Fundamentals, Pearson Education, 2011.
3. Linda Foulkes, Learn Microsoft Office 2019: A comprehensive guide to getting started with Word, PowerPoint, Excel, Access, and Outlook, Packet Publishing Limited, 2020.
4. Bittu Kumar, Mastering MS Office: Concise Handbook with Screenshots, V&S Publishers, 2017.

II Semester

Course Code: G505DC2.2	Course Title: Data Structures using C
Course Credits: 4	Hours of Teaching/Week: 4
Total Contact Hours: 52	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hours

Course Outcomes (COs)

After completing this course satisfactorily, a student will be able to:

- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
- Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs
- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
- Demonstrate different methods for traversing trees
- Compare alternative implementations of data structures with respect to performance
- Describe the concept of recursion, give examples of its use
- Discuss the computational efficiency of the principal algorithms for sorting and searching

Course Content

Content	Hours
Unit - 1	
Introduction to Data Structures: Definition, Need for Data Structures, Types of Data Structures. Linear Data Structures: Arrays - Definition, Declaration and storage of one- and two-dimensional arrays. Sparse matrices. Recursion: Definition; Types of recursion; Recursion Technique Examples - Fibonacci numbers, GCD, Binomial coefficient nC_r , Comparison between iterative and recursive functions. Sorting: Sorting – Selection sort, Bubble sort, Quick sort, Insertion sort; Comparison of different sorting techniques.	13
Unit - 2	
Searching: Introduction, Linear search, Binary Search, Comparison of different searching techniques. Dynamic memory allocation: Static and Dynamic memory allocation; Memory allocation and deallocation functions - <i>malloc</i> , <i>calloc</i> , <i>realloc</i> and <i>free</i> . Linked List: Introduction, characteristics, types of linked lists, Representation of singly linked list in memory, Singly linked list – Operations, algorithms, Representation of polynomials using linked lists. Circular linked list – Operations, Doubly linked list - operations. Memory allocations.	13
Unit - 3	
Stacks – Array representation of stacks, Linked representation of stacks, operations, Applications of stacks Recursion, Implementation of recursive procedure by stack (factorial function and Fibonacci sequence).	13

<p>Arithmetic Expressions: Prefix, infix and postfix notation, infix to postfix conversion, evaluation of postfix expression.</p> <p>Queues: Array representation of queue, Linked representation of queue, Types of queues- Simple queue, circular queue, double-ended queue, priority queue, operations on queues.</p>	
Unit - 4	
<p>Trees: Definition; Tree terminologies – node, root node, parent node, ancestors of a node, siblings, terminal and non-terminal nodes, degree of a node, level, edge, path, depth;</p> <p>Binary tree: Types of binary trees - strict binary tree, complete binary tree, binary search tree. Array representation of binary tree. Traversal of binary tree; <i>preorder</i>, <i>inorder</i> and <i>postorder</i> traversal; Construction of a binary tree when inorder and pre/postorder traversals are given.</p> <p>Graphs: Terminologies, Matrix representation of graphs, Traversals: Breadth First Search and Depth first search.</p>	13
<p>Text Books</p> <ol style="list-style-type: none"> 1. Sartaj Sahni: Fundamentals of Data Structures. 2. YedidyahLang sam, Moshe J. Augenstein and Aaron M. Tenenbaum, Data Structures Using C and C++, 2ndEdition, PHI Publication 3. Seymour Lipschutz, Schaum’s Outlines Series, Data Structures with C, Tata McGraw Hill 2011 <p>References</p> <ol style="list-style-type: none"> 1. Kamathane: Introduction to Data structures (Pearson Education) 2. Y. Kanitkar: Data Structures Using C (BPB) 3. Kottur: Data Structure Using C 4. Padma Reddy: Data Structure Using C 5. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007) 	

Course Code: G505DC2.2P	Course Title: Data Structures Lab
Course Credits: 2	Hours of Teaching/Week: 4
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 3 Hours

Programming Lab

Part A

1. Program to compute power of a number using a recursive function.
2. Program to compute the value of ${}^n C_r$ using a recursive function to find factorial function.
3. Program to implement dynamic array, find smallest and largest element of the array.
4. Program to read the names of cities and arrange them alphabetically.
5. Program to sort the given list using selection sort technique.
6. Program to sort the given list using bubble sort technique.
7. Program to sort the given list using quick sort technique.
8. Program to sort the given list using insertion sort technique.
9. Program to search an element using linear search technique.
10. Program to search an element using recursive binary search technique.

Part B

1. Program to implement queue using arrays.
2. Program to implement stack using arrays.
3. Program to implement stack using linked list.
4. Write a Program for converting an Infix Expression to Postfix Expression. Program should support both parenthesized and free parenthesized expressions with the following operators: +, -, *, /, % (Remainder), ^ (Power) and alphanumeric operands.
5. Program to evaluate a postfix expression.
6. Program to implement all operations on a sorted singly linked list.
7. Program to implement queue using linked list.
8. Program to implement circular queue using array.
9. Write a menu driven program for the following operations on Binary Search Tree (BST) of integers:
 - (a) Create a BST of 'n' Integers
 - (b) Traverse the BST in Inorder, Preorder and Postorder
 - (c) Search the BST for a given element and report the appropriate message
10. Program for the following operations on a graph (G) of cities:
 - (a) Create a graph of N cities using Adjacency Matrix.
 - (b) Print all the nodes reachable from a given starting node in a digraph using BFS method.

Evaluation Scheme for Practical Examination:

Assessment Criteria	Marks
Writing the Program	7
Execution and Formatting	8
Writing the Program	8
Execution and Formatting	12
Practical Records	10
Viva	5
Total (converted to 25)	50

Course Code: G5050E2.2	Course Title: Web Designing
Course Credits: 3	Hours of Teaching/Week:3
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hours

Course Outcomes (COs)

After completing this course satisfactorily, a student will be able to:

- Understand various Internet related terminologies
- Explain features and evolution of Internet
- Explain the use of search engines
- Know the use of different tags available in HTML
- Design web pages using HTML5, CSS3, XML and XHTML
- Implement websites using linked web pages.

Course Content

Content	Hours
Unit - 1	
The Internet: Introduction, Evolution, basic internet terms, Getting connect to internet, Internet applications, Data over the internet. Internet tools: Web browser, Web browser features, Internet Explorer environment, Electronic mail, Email address structure, checking email, sending email, email attachment, How email works, advantages and disadvantages of email. Search Engines: Searching an internet, refining the search, Instant messaging, Features of messengers.	12
Unit - 2	
Overview of HTML5 -Exploring new features of HTML5, Structuring an HTML Document, Creating and saving HTML document, Viewing an HTML document. Fundamentals of HTML -Understanding Elements, Root elements, Metadata elements, Style element, Section element, Header and Footer element, Address element, Basic HTML data types, Data types defined by RFC and IANA Documentation. Working with Text: Formatting Text with HTML Elements, Defining MARK element, Defining STRONG element, Defining CODE element, Defining SMALL element. Organizing Text in HTML: Arranging text, Displaying Lists.	10
Unit - 3	
Working with Links and URLs- Exploring the Hyperlinks, Exploring the URL, Exploring Link Relations. Creating Tables- Understanding Tables, Describing the table element. Working with Images, Colors and Canvas - Inserting images in a web page, Exploring Colors, Introducing Canvas Working with Forms: Exploring Form element, Exploring types of the INPUT element, Exploring the BUTTON element, Exploring the Multiple-Choice elements, Exploring TEXTAREA and LABEL elements.	10

Working with Frames: <FRAMESET>, <FRAME> tag with attributes.	
Unit - 4	
Overview of CSS3- Understanding the syntax of CSS, Exploring CSS Selectors, Inserting CSS in an HTML document.	10
Background and Color Gradients in CSS: Exploring Background of a Web Page, Exploring Color Properties, Exploring Gradient Properties, Exploring Font properties.	
Working with Basics of XML- Exploring XML, Comparing XML with HTML, Describing the Structure of an XML document.	

Text Books:

1. ITL Education Solution Limited, Introduction to Information Technology, Pearson Education, 2012
2. DT Editorial Services, HTML 5 Black Book (Covers CSS3, JavaScript, XML, XHTML, AJAX, PHP, jQuery), Second Edition, Dreamtech Publisher, 2016

References:

1. Laura Lemay & Rafe Colburn, Mastering Html, CSS & JavaScript, Web Publishing, 2016
2. Firuza Aibara, HTML 5 for Beginners, 2012
3. Glenn Johnson, Training Guide – Programming in HTML5 with JavaScript and CSS3 (Microsoft Press Training Guide), 2013

III Semester

Course Code: G505DC3.3	Course Title: Object Oriented Programming Concepts and Programming in JAVA
Course Credits: 4	Hours of Teaching/Week: 4
Total Contact Hours: 52	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 3 Hours

<p>Course Objectives</p> <ul style="list-style-type: none"> •To learn the concepts of Object-Oriented Programming. •To learn the Object-oriented programming using Java. <p>Course outcomes:</p> <p>Upon successful completion of the course the student will be able to:</p> <ul style="list-style-type: none"> •Understand the concepts of OOP and Java fundamentals. •Write the Java programs using the concepts of inheritance, interfaces, packages, multithreading and applets. 	
Content	Hours
Unit - 1	
<p>Fundamentals of Object-Oriented Programming: Introduction, Object-oriented paradigm, Basic concepts of OOP, Benefits of OOP, Applications of OOP.</p> <p>Java Fundamentals: Java Evolution, Overview of Java language. Comparison of C and Java. Data Types and Operators: Java's Primitive Types, Literals, Variables, The Scope and Lifetime of variables, Type Conversion in Assignments, Casting Incompatible Types.</p> <p>Control Statements: Additional features of for loop, labelled loops.</p>	13
Unit - 2	
<p>classes, objects and methods: defining a class, fields and method declaration, creating objects, accessing class members, constructors, method overloading, static members, nesting of methods.</p> <p>Inheritance: extending a class overriding methods, final variables and methods, final classes, finalizer methods, abstract methods and classes, visibility control.</p> <p>Arrays strings and Vectors: one dimensional array, creating Arrays, Two dimensional arrays, Strings, vectors, wrapper classes</p>	13
Unit-3	
<p>Packages and Interfaces: Packages, Packages and Member Access, Understanding Protected members, Importing packages, Java's standard packages, Interfaces, Implementing and extending Interfaces.</p> <p>Multithreaded Programming: creating a thread, extending a thread class, life cycle of a thread, thread priority, using thread methods, synchronization, implementing runnable interfaces.</p> <p>Managing Errors and Exception: types of error, syntax of exception handling, multiple catch statements, using finally statement, throwing our own exceptions.</p>	13
Unit-4	
<p>Applets Programming: Building Applet Code, Applet lifecycle, creating an executable Applet, applet tag, passing parameters to Applet, event handling</p> <p>graphics programming: graphics class, lines and rectangles, circle, ellipse, arcs, polygons, introduction to AWT packages.</p>	13

Text Books:

1. E Balagurusamy, **Programming With Java: A Primer, 5th edition** Tata McGraw Hill Education Private Limited, 2010.

Reference books:

1. Herbert Schildt, **Java: A Beginner's Guide, 5th Edition** Tata McGraw Hill Education Private Limited,
2. Junaid Khateeb and Dr. G T Thampi, **Computer Programming in Java**, Dreamtech, 2011
3. Herbert Schildt, **The Complete reference Java**, Seventh edition, Tata McGraw Hill Publishing

Course Code: G505DC3.3P	Course Title: JAVA lab
Course Credits: 2	Hours of Teaching/Week: 4
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hours

PART-A

- Program to accept student name and marks in three subjects. Find the total marks, average and grade (depending on the average marks).
- A menu driven program to input two integers & an operator to perform basic arithmetic operations (+, -, *, /) using switch case structure.
- Program, which reads two numbers having same number of digits. The program outputs the sum of product of corresponding digits. (Hint Input 327 and 539 output $3 \times 5 + 2 \times 3 + 7 \times 9 = 84$)
- Program to input Start and End limits and print all Fibonacci numbers between the ranges. (Use for loop)
- Define a class named Pay with data members String name, double salary, double da, double hra, double pf, double grossSal, double netSal and methods: Pay(String n, double s) - Parameterized constructor to initialize the data members, void calculate() - to calculate the following salary components, and void display() - to display the employee name, salary and all salary components.
 Dearness Allowance = 15% of salary
 House Rent Allowance = 10% of salary
 Provident Fund = 12% of salary
 Gross Salary = Salary + Dearness Allowance + House Rent Allowance
 Net Salary = Gross Salary - Provident Fund
 Write a main method to create object of the class and call the methods to compute and display the salary details.
- Program to create a class DISTANCE with the data members feet and inches. Use a constructor to read the data and a member function Sum () to add two distances by using objects as method arguments and show the result. (Input and output of inches should be less than 12.)
- Program to check whether the given array is Mirror Inverse or not.
- Program to create a class "Matrix" that would contain integer values having varied numbers of columns for each row. Print row-wise sum.
- Program to extract portion of character string and print extracted string. Assume that 'n' characters extracted starting from mth character position.
- Program to add, remove and display elements of a Vector.

PART B

- Create a class named 'Member' having data members: Name, Age, PhoneNumber, Place and Salary. It also has a method named 'printSalary' which prints the salary of the members. Two classes 'Employee' and 'Manager' inherit the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.
- Program to implement the following class hierarchy: Student: id, name StudentExam (derived from Student): Marks of 3 subjects, total marks StudentResult (derived from StudentExam) : percentage, grade
 Define appropriate methods to accept and calculate grade based on existing criteria and display details of N students
- Program to calculate marks of a student using multiple inheritance implemented through interface. Class Student with data members rollNo, name, String cls and methods to set and put data. Create another class test extended by class Student with data members mark1, mark2, mark3 and methods to set and put data. Create interface sports with members sportsWt = 5 and putWt(). Now let the class results extends class test and implements interface sports. Write a Java program to read required data and display details in a neat format.
- Program to create an abstract class named shape that contains two integers and an empty method named print Area(). Provide three classes named Rectangle, Triangle and Ellipse such that each one of the classes extends the class shape. Each one of the class contains only the method print Area() that print the area of the given shape.

Evaluation Scheme for Practical Examination:

Assessment Criteria		Marks
Program-1 from Part A	Writing the Program	7
	Execution and Formatting	8
Program-2 from Part B	Writing the Program	8
	Execution and Formatting	12
Practical Records		10
Viva		5
Total (converted to 25)		50

III Semester		
Course Code: CSOE2/CAOE2	Course Title: IoT	
Course Credits: 3	Hours of Teaching/Week:3	
Total Contact Hours: 42	Formative Assessment Marks: 40	
Exam Marks: 60	Exam Duration: 3 Hours	
Course Objectives		
OBJECTIVES:		
1. To introduce the concept of “Internet of Things “to the students.		
2. To understand the basic ecosystems and landscape in IoT.		
3. To understand the Realtime use cases in IoT		
4. To know the different domains, where IoT plays a crucial role.		
Course outcomes		
<ul style="list-style-type: none"> • To become familiar with the basic concepts of IoT. • To become familiar with IoT access techniques 		
Content	Hours	
Unit - 1		
Unit1	13	
Introduction to IoT: What is IoT, IoT architecture, Characteristics of IOT systems, Prevalent IoT architectures applications, Overview of different technologies involved for IoT realization		
Unit - 2		
History of IoT: The transition from mainframes and personal computing, Planet lab and origins of distributed computing; Robotics, AI and Cyber Computing Infrastructure; M2Mcommunications; P2P networks; Universal identification and RFID; Autonomic computing, Pervasivecomputing , Ubiquitouscomputing; WirelessSensorNetworks, TheemergenceofIoT.	13	
Unit-3		
IOT state of the art: The IoT ecosystem and landscape; IOT business models and its usage invarious domains; Technology Enablers for IOT – Mobility, Analytics, Cloud and Social Media;IOTplatforms; Security;Testmethodologies; Regulations andRisks.	13	
Unit-4		
IoT Characteristics and use cases: Consumer and enterprise use cases IoT DOMAINS: Smart Home, Smart Buildings, smart cities, IoT in telecommunications, Smart manufacturing, IoT in environment monitoring, smart vehicles, IoT in healthcare, smart farming, IoTin enterprises, smart transportation,smart energy,smart retail and logistics	13	
1. VijayMadiseti,ArshdeepBahga “Internet of things, A hands-onapproach”2014 2. Jean-Philippe Vasseur&Adam Dunkels“Interconnecting smart objects withIP”,Morgan KaufmannPublishers,2010 3. CunoPfister, “Getting Started with the Internet of Things” , Maker MediaInc,2011 4. Adrian Mcewen and Hakim, “Designing the Internet of Things”, Wileypublication,2013		

TEXT BOOK :

1. Arshadeep Bhaga and Vijaya Madiseti, Internet of Things, A Hands an Approach, Universities Press, 2014.

2. Raj Kamal, *Internet of Things: Architecture and Design Principles*, Mc Graw Hill Education .

Reference Books:

1. Rob Barton, Gonzalo Salgueiro, David Hanes, IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things, Cisco Press,2017.

IV Semester		
Course Code: DSC-4	Course Title: Data Base Management System	
Course Credits: 4	Hours of Teaching/Week: 4	
Total Contact Hours: 52	Formative Assessment Marks: 40	
Exam Marks: 60	Exam Duration: 3 Hours	
Course Objectives		
<ul style="list-style-type: none"> • To Understand the basic concepts and the applications of database systems • To Master the basics of SQL and construct queries using SQL • To understand the relational database design principles 		
Course outcomes		
<ul style="list-style-type: none"> • To become familiar with the basic issues of transaction processing and concurrency control • To become familiar with database storage structures and access techniques 		
Content	Hours	
Unit - 1		
Database System Concepts and Architecture History of Database Systems, Characteristics of the Database Approach, Actors on the Scene, Workers behind the Scene, Advantages of Using the DBMS Approach, Database Systems versus File Systems, Data Abstraction, Three-Schema Architecture Data independence, Schemas and Instances, Data models, Database Languages, Database Users, DBA, Structure of Database Systems. The database system environment, Centralized and Client/Server Architecture for DBMS, Classification of DBMS, Entity types, attributes, keys, relationships, relationship types, roles and structural constraints, Weak entity sets, Database Abstractions- Generalization, Aggregation, Data Modeling using E-R Models.	13	
Unit - 2		
Relational model: Basic Concepts of relational data model, Relational model constraints and relational database schemas-Domain Constraints, Key Constraints, Relational Database Schema, Basic Relational algebra operations-SELECT, PROJECT, Set Operations, Cartesian Product, Renaming, Join Operations, Division. Design theory of Relational Database: Introduction to Relational database design, Semantics of Attributes, Problems caused by Redundancy-Anomalies, Functional dependency- Uses of Functional Dependency, Diagrammatic way of showing FDs, Closure Set Properties, Normal forms (1NF, 2NF,3NF and BCNF)	13	
Unit-3		
MySQL Introduction to Transaction Control Language (TCL) – Commit, Rollback, Savepoint. Creating and Using New Users, Granting, Revoking, and viewing User Privileges. Creating Database, adding tables, adding records - Single row or multiple rows at a time, Executing queries. MySQL Datatypes, The ENUM type, MySQL Operators – Arithmetic, Comparison, Logical operators, Aggregate functions, String manipulation, Date and time manipulation functions, Selecting databases for use, deleting databases, LIKE clause, The NULL and NOT NULL Modifiers, The DEFAULT Modifiers, The AUTO_INCREMENT Modifier, The UNIQUE Modifier, Primary Keys, Foreign Keys, Modifying tables – Alter, Deleting tables, Inserting, Updating, Deleting Records, Retrieving records, Aliasing table and column names, Limiting query results-limit, Sorting Query Results-Order by, Grouping Query results- Group By, Having, Joins- Inner Join ,Left Join, Right Join, Self-Join, Cross Join, Nested Queries, The IN, NOT IN,	13	

and BETWEEN Clause, sub queries, On DELETE CASCADE.		
Unit-4		
Views in MySQL What are Views in MySQL. Advantages of Views, Disadvantages of VIEWS, Creating Views, MYSQL Updatable Views, MySQL Views with CHECK OPTION. MySQL Stored Procedures Stored Program, three major types of MySQL stored programs, Drawbacks of using stored procedures. Language Fundamentals- Variables, Assigning Values to Variables, Parameters, Conditional Statements, And Looping statements (Iterative Programming), Using SELECT Statements with an INTO Clause, Creating and Using Cursors, fetching a Single Row from a Cursor, Fetching an Entire Result Set, Creating Stored Functions, MySQL Error handling in stored procedure.	13	

Text Book:

1. Elmasri and Navathe, Fundamentals of Database Systems, Pearson Education Asia Publication, 4th edition. (Unit 1)
2. S.Nanda Gopalan, Data base Management Systems with oracle 9i and VB 6.0, 4th Edition, Sapna Book House Bangalore. (Unit 2)
3. Vikram Vaswani “The complete Reference MySQL “Tata McGraw-Hill Edition 2004, Eleventh reprint 2009. (Unit 3)
4. Guy Harrison with Steven Feuerstein “MySQL Stored Procedure Programming”. O’REILLY (Unit 4)

Reference Book :

1. Seyed M.M. “Saied” Tahaghoghi and Hugh E. Williams “Learning MySQL” O’REILLY
2. Silberschatz and Korth , Database System Concepts, McGrawHill Publication
3. Ivan Bayross, Commercial Application Development using Oracle D2k, BPB Publications.

Course Code: G505DSC4.4	Course Title: RDBMS LAB
Course Credits: 2	Hours of Teaching/Week: 4
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hours

LIST OF PROGRAMS - PART A (MYSQL Queries)

PART-A

1. Create a database **Movie_Industry** under which Create a table **Movie** with the following columns:

Table name: Movie

COLUMN	DATA TYPE	CONSTRAINTS
ID	Int	Auto_Increment
Movie_name	Text	UNIQUE
Description	ENUM	Must be Great 3D,Fiction, Boring, Fantasy, Interesting
Rating	INT	NOT NULL

Insert multiple records at once to the table

Alter the table to add the columns Director Name and Production

Display all the details of the movies.

QUERIES:

1. Display all the movie names in alphabetical order belong to Fiction.
2. Display all not boring movies, names and ratings information (use IN predicate).
3. Display the movie name and the description directed by Shankar.

4. Count the number of movies under Interesting journal.
5. Count how many 3D movies are directed by Ang Lee.
6. Display the movie names and the production where the director's name contains 'Jackson'
7. Delete all the boring movies.
8. Rename the table name as cinema.
9. Display the movie name and director name which has 3rd highest rating.
10. Assign update privilege only on rating column on the table cinema
11. to Jhon.

2. Create a database COMPANY under which Create a table FACULTY with the following columns

COLUMN	DATA TYPE	CONSTRAINTS
FID	INT	PRIMARY KEY
FNAME	VARCHAR2	NOT NULL
DEPT	VARCHAR2	NOT NULL
BASIC	DOUBLE	GREATER THAN 10000

Insert minimum five records to the table

Alter the table to add the columns DA, HRA, PF, Net pay, IT. Calculate the DA, HRA, PF, IT, NETPAY as follows: DA=80% of BASIC, if BASIC<20000 else 70% of BASIC HRA=10% of BASIC

NETPAY=BASIC+DA+HRA

PF=2% of NETPAY and IT=5% of NETPAY

QUERIES:

1. Display all the available databases in MySQL.
2. Display the details of all faculty details.
3. Display the faculty names of computer science department.
4. Display the faculty names in alphabetical order.
5. List the highest net pay faculty in each department.
6. Display the details of all faculty whose net pay between 25000 and 35000
7. Display the name and basic of employees whose name contains 'Kumar'
8. Count the number of employees in each department
9. Create a new MySQL user account
10. Delete all employees from home science department

3. Create a table EMP_MASTER under COMPANY database with the following columns:

COLUMN	DATA TYPE	CONSTRAINTS
ENO	INT	Primary Key
ENAME	VARCHAR	NOT NULL
DEPTNO	INT	NOT NULL
JOB_TITLE	ENUM	Must be 'Sales Rep', 'Marketing', 'Administrative Assistant'
SALARY	DOUBLE	NOT NULL
COMMISSION	DOUBLE	
JOIN_DATE	DATE	NOT NULL

QUERIES:

1. Show all the tables available in company database.
2. Modify eno column as primary key.
3. Display the employee's name, department and their job title with a second highest salary.
4. List the names of employees whose names are five character long
5. Display the employee details and sort the results as per job title 'Marketing', 'Sales Rep', 'Administrative

Assistant'.

6. Find all the department that have at least 2 Sales Rep.
7. Display the no of employees joined in the year 2014.
8. Display the first 4 records of the emp_master table.
9. who don't earn commission ,10% increase in salary and commission that is 15% of their salary?
10. Assign SELECT and INSERT privileges on the table emp_master to Harry.

4. Create the following tables under College database with the following columns:

Table Name: Faculty

COLUMN	DATA TYPE	CONSTRAINTS
Fid	Int	Primary Key
Fname	Varchar	NOT NULL
Qualification	Varchar	NOT NULL
DeptId	Int	Foreign key References Department (DeptId)

)

Table Name : Department

COLUMN	DATA TYPE	CONSTRAINTS
DeptId	Int	Primary Key
Dname	Varchar	NOT NULL

QUERIES:

1. Display the department name where there is no Ph.D qualified faculty.
2. Perform cross join between Faculty and Department tables the result should contain department no 1.
3. Find department name of each faculty (use inner join)
4. Create a view that contain Faculty Name and Corresponding Department Name.
5. Display how many faculties are there in each department.
6. Find the names of the faculty working in IT Department.
7. Display the department name where there is no Faculty enrolled.
8. Perform left join between department and faculty tables display the faculty name, department name.

5. Table Name: Sailors

COLUMN	DATA TYPE	CONSTRAINTS
Sid	Int	Primary Key
Sname	Varchar	NOT NULL
Rating	Int	NOT NULL
Age	Int	NOT NULL

Table Name: Boats

COLUMN	DATA TYPE	CONSTRAINTS
Bid	Int	Primary Key
Bname	Varchar	NOT NULL
Color	Varchar	NOT NULL

Table Name: Reserves

COLUMN DATA CONSTRAINTS

TYPE

Sid	Int	Foreign key References Sailors(Sid)
Bid	Int	Foreign key References Boats(Bid)
Day	Date	NOT NULL

QUERIES:

1. Find all information of sailors who have reserved boat number 101.
2. Find the name of boat reserved by Brutus.
3. Find the names of sailors who have reserved a red boat, and list in the order of age.
4. Find the names of sailors who have reserved at least one boat.
5. Find the ids and names of sailors who have reserved two different boats on the same day.
6. Find the name and the age of the youngest sailor.
7. Perform left outer join between sailors and reservation tables display the sid, sname, bid in ascending order.
8. Find the average age of sailors for each rating level.

6. Create the following tables under LIBRARY database with the following columns:**Table Name: Book**

COLUMN	DATA TYPE	CONSTRAINTS
Book_id	Int	Primary Key
Title	Varchar	NOT NULL, UNIQUE
Publisher_Name	Varchar	NOT NULL
Pub_Year	Year	NOT NULL

Table Name: Book_Lending

(Book_id, Card_No)-is Composite Primary Key.

COLUMN	DATA TYPE	CONSTRAINTS
Book_id	Int	Foreign key References Book (Book_id) On Delete Cascade
Card_NO	Int	NOT NULL
Date_Out	Date	NOT NULL
Due_Date	Date	NOT NULL

Table Name: Book_Copies

(Book_id, Branch_name)-is Composite Primary Key.

COLUMN	DATA TYPE	CONSTRAINTS
Book_id	Int	Foreign key References Book (Book_id) On Delete Cascade
Branch_name	Varchar	NOT NULL
No_of_Copies	Int	NOT NULL

QUERIES:

1. Display the particulars of borrowers and the book name who have borrowed books from Jan 2021 to March 2021.
2. Create a view of all books and its number of copies that are available in the library.
3. Display branch wise available books name and total no of copy available and list in the order of branch

name.

4. Display the particulars of borrowers and the book name who have borrowed more than 3 books, but from Feb 2021 to April 2021.
5. Create a view table, which consist of book title, publication and year of publication. (year wise latest books first)
6. Display how many books are available under the 'PEARSON' Publication.
7. Display the branch name which has less than 35 books as a whole.
8. Delete Book_id 1 from book table and show that all the relevant data of the same bookid got deleted from book_lending, book_copies tables;

Scheme of Assessment for Theory Examination

Question Pattern		Marks
Part – A		
1. Answer any SIX sub-questions (6×2=12)		12
Sub-question	Unit	
a, b	1	
c, d	2	
e, f	3	
g, h	4	
Part – B (Answer any ONE full question from each unit – 12 marks each) (Combinations of sub-questions of 3 to 6 marks)		
Unit-1		12
2.		
3.		
Unit-2		12
4.		
5.		
Unit-3		12
6.		
7.		
Unit-4		12
8.		
9.		
Total		60