

Proposed Syllabus

For

B.Sc (Gen.) in Computer Science

Submitted

To

University of Gour Banga

Under

Choice Based Credit System (CBCS)

[With effect from the Session 2019- 20]

Semester	Discipline Core (DC)	Discipline Specific Elective (DSE)	Ability Enhancement Compulsory (AEC)	Skill Enhancement (SEC)	Credits	Marks
SEM-I	DC1: Computer Fundamentals	--	ENVS	--	20	200
SEM-II	DC2: (a) Introduction to C Programming (b) C Programming Lab	--	Communicative English/Communicative Bengali/ Modern Indian Language	--	20	200
SEM-III	DC3: Data Structure using C	--	--	SEC-1: Digital System Design	20	200
SEM-IV	DC4: (a) Database Management System (b) DBMS with MySql	--	--	SEC-2: Microprocessor 8085	20	200
SEM-V	---	DSE-1. E1: Data Communication & Networking Or E2: System Analysis & Design	--	SEC-3: Introduction to Python Programming	20	200
SEM-VI	---	DSE-2 E1. (a) Operating System (b)Operating System Lab Or DSE-2. E2. (a) Object Oriented Programming using C++ (b) Object Oriented Programming Lab using C++	--	SEC-4: Web Design	20	200
Total	--	--	--	--	120	1200

Semester – I

DC1: Computer Fundamentals: 60 Hours

Introduction to Computer and Problem Solving: Information and Data

Hardware: CPU, Primary and Secondary storage, Cache Memory, I/O devices, Bus structure, BIOS

Software: System and Application.

Introduction to Programming Languages: Machine Language, Assembly Language, High Level Language.

Problem Solving: Flow Charts and Pseudo codes.

System Software: Classifications- Operating Systems (OS); Translators – Compilers and Interpreters (concepts only), Pre-processors, Assemblers, Loaders, Linkers other utilities.

Virus: Concept, Detection and Protection

Multimedia: Basic Concept, associated hardware and software

Object Oriented Paradigm: Basic characteristics, Definition, Brief comparison with other types of programming paradigms.

Text/Reference Books:

1. Digital Circuits and Design, Salivahan, Vikas.
2. Computer Fundamentals, Anita Goel, Pearson Education.
3. Introduction to Computer Science, P.K.Sinha, P.Sinha, BPB Publication.

Semester – II

DC2:(a)Introduction to C Programming: 60 Hours

Introduction: Basic Structure, Algorithms, Flowcharts, Structured programming constructs.

C Programming elements: Character sets, Keywords, Constants, Variables, Data Types, Operators- Arithmetic, Relational, Logical and Assignment; Increment and Decrement and Conditional Operator, Precedence and Associations; Expressions, type casting. Comments, Functions, Storage Classes, Bit manipulation, Input and output.

C Pre-processor: File inclusion, Macro substitution.

Statements: Assignment, Control statements- if, if else, switch, break, continue, goto, Loops- while, do_while, for.

Functions: Argument passing, return statement, return values and their types, recursion

Arrays: String handling with arrays, String handling functions.

Pointers: Definition and initialization, Pointer arithmetic, Pointers and arrays, String functions and manipulation, Dynamic storage allocation.

User defined Data types: Structures. Structure arrays, Pointers to Functions and Structures, Unions

File Access: Opening, Closing, I/O operations.

DC2: (b) C Programming Lab: 40 Hours

Some sample examples are given below. More problems can be included related to the theory.

Use open source C compiler (GCC).

1. WAP to print the sum and product of digits of an integer.

2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series, $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
4. WAP to compute the sum of the first n terms of the following series, $S = 1 - 2 + 3 - 4 + 5 - \dots$
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. WAP to print a triangle of stars as follows (take number of lines from user):


```

*
***
*****
*****
*****
*****

```
10. WAP to perform following actions on an array entered by the user :
 - i) Print the even-valued elements
 - ii) Print the odd-valued elements
 - iii) Calculate and print the sum and average of the elements of array
 - iv) Print the maximum and minimum element of array
 - v) Remove the duplicates from the array
 - vi) Print the array in reverse order. The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.
11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.
12. Write a program that swaps two numbers using pointers.
13. Write a program in which a function is passed address of two variables and then alter its contents.
14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.
15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.
16. Write a menu driven program to perform following operations on strings:
 - a) Show address of each character in string
 - b) Concatenate two strings without using strcat function.
 - c) Concatenate two strings using strcat function.
 - d) Compare two strings
 - e) Calculate length of the string (use pointers)
 - f) Convert all lowercase characters to uppercase
 - g) Convert all uppercase characters to lowercase
 - h) Calculate number of vowels
 - i) Reverse the string
17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.
18. WAP to display Fibonacci series (i) using recursion, (ii) using iteration.

19. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration.
20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.
21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
a) Sum b) Difference c) Product d) Transpose
22. Copy the contents of one text file to another file, after removing all whitespaces.
23. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
24. Write a program that will read 10 integers from user and store them in an array.

Text/ Reference Books:

1. Programming with C, Byron S. Gottfried, McGraw Hill.
2. The C Programming Language, Kernighan and Dennis, PHI.
3. The Complete reference C, Herbert Schildt, McGraw Hill.
4. Let Us C, Kanitkar, BPB Publication.
5. Programming in ANSI C, Balaguruswamy, McGraw Hill.
6. Programming Languages, Allen B. Tucker, Tata McGraw Hill.

Semester – III

DC3: Data Structure using C : 60 Hours

Introduction: Algorithms, ADT.

Arrays: One dimensional and Two Dimensional Arrays, Row Major and Column Major ordering, different operations on array.

Linked List: Singly and Doubly Linked List; Operations Like Insertion, Deletion, Searching etc.

Stacks and Queues: Concepts of Stack and Queue; Insertion and Deletion of Elements; Array and Linked Representation: Prefix, Infix and Postfix Notation; Postfix Expression Evaluation, Infix to Postfix conversion.

Searching: Sequential, Binary Search Techniques.

Trees: Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Trees), Binary Search Tree;

Sorting: Bubble Sort, Selection Sort, Insertion Sort.

Text/ Reference Books:

1. Fundamentals of Data Structures in C, Ellis Horowitz, Sartaj Sahni, Susan Anderson-Freed, Silicon Pr.
2. Data Structures and algorithm in C, Adam Drozdek, Cengage Learning.
3. Data Structures, Algorithms and applications in C++, Sartaj Sahni, Universities Press.
4. Data Structures Using C and C++, Aaron M. Tanenbaum, Moshe J. Augenstein, Yedidyah Langsam, PHI.
5. Classic Data Structures, Debasis Samanta, PHI

SEC-1: Digital System Design: 60 Hours

Number Systems and Codes: Number representation: Weighted Codes, Non-weighted codes, Positional, Binary, Octal, Hexadecimal, Binary Coded Decimal (BCD), Conversion of bases. Complement notions. Binary Arithmetic, Binary Codes: Gray, Alphanumeric, ASCII, EBCDIC; Single Error-Detecting and Correcting Codes, Hamming Codes, Fixed point, Floating point representation.

Boolean Algebra: Fundamentals of Boolean Algebra, Switches and Inverters, Functionally Complete Gates (AND, OR, NOT), NAND, NOR, Switching function and Boolean Function. De Morgan's Theorem, Min-term, Max term, Truth tables and minimization of switching function upto four variables, Algebraic and K-map method.

Digital Electronics: Combinational Circuits: Realization of Boolean expression using AND, OR and NOT Gate, Half and Full Adder/ Subtractor, Data selectors/multiplexers –function realization, Decoders: function realization, De-multiplexer, Encoder, Priority Encoder, Parity bit Generator/checker, Code Converters.

Sequential Circuits: Model of Sequential computing, Difference between Combinational and Sequential circuit, RS-Latch: using NAND and NOR Gates, Flip Flops - SR, JK, D, T, Level Trigger and Edge Trigger, Excitation Functions of each flipflops, Application of Flip-flops: Asynchronous Counter (UP/DOWN) upto 3 bit, Registers: Registers with parallel load, Shift Registers.

Text/Reference Books:

1. Digital Circuits, Vol - I & II, D. Ray Chaudhuri, Platinum Publishers.
2. Digital Systems - Principle & Applications, Tocci & Widmer, EEE.
3. Digital Logic & State Machine Design, Comer, Oxford.
4. Digital Principle & Applications, Malvino & Leach, McGraw Hill.
5. Digital Design, Mano, PHI.
6. Digital Integrated Electronics- H.Taub & D.Shilling, Mc Graw Hill.
7. Digital Circuits and Design, Salivahan, Vikas.

Semester – IV

DC4: a) Database Management System: 60 Hours

Introduction: DBMS, Advantages of DBMS; Applications of DBMS; Layered Architecture of Database, Data Independence; Data Models; Schemas and Instances; Database Languages;

ER Model: Entity, Attributes and Relationship; Structural Constraints; Keys; ER Diagram of Some Example Database; Weak Entity Set; Symbolic Conventions; Specialization and Generalization; Constraints of Specialization and Generalization; Aggregation.

Relational Model: Concepts of Relational Model; Relational Algebra;

Integrity Constraints: Domain Constraints, Referential Integrity.

Relational Database Design: Problems of Un-Normalized Database; Functional Dependencies, Derivation Rules, Closure of FD Set, Decomposition to 1NF, 2NF, 3NF and BCNF Using FDs.

DC4: b) DBMS with MySql: 40 hours

SQL: Basic Structure, Data Definition, Constraints and Schema Changes; Basic SQL Queries (Selection, Insertion, Deletion, Update); Order by Clause; Complex Queries, Aggregate Function and Group by Clause; Nested Sub Queries; Correlated Sub Queries; Joined Relations; Set Comparisons.

Text/ Reference Books:

1. Fundamentals of Database Systems, R. Elmasri, S.B. Navathe, Pearson Education.
2. Database Management Systems, R. Ramakrishanan, J. Gehrke, McGraw-Hill.
3. Database System Concepts 6th Edition, A. Silberschatz, H.F. Korth, S. Sudarshan, McGraw Hill.
4. Database Systems Models, Languages, Design and application Programming, R. Elmasri, S.B. Navathe, Pearson Education.

SEC-2: Microprocessor 8085: 60 Hours

Introduction to Microcomputer based system: Microprocessor and Microcontrollers and their advantages and disadvantages.

Microprocessor Architecture and Memory Interfacing: Basic Architecture of Microprocessor 8085 and explanation of each block, Microprocessor 8085 pin out and signals, Addressing modes, Instruction Formats, Instruction Cycle, Clock Cycle, Multiplexed Address Data Bus, Control and Status signals, Microprocessor and Bus Timing, De-multiplexing of Address Data Bus, Generation of Control Signals for I/O and Memory, Basic concepts in Memory Interfacing, Address Decoding and memory Addresses.

Interfacing I/O Devices: Basic Interfacing concepts, Peripheral I/O instructions (I/O mapped I/O),

Programming 8085: Instruction Set of 8085, Different Programming Techniques, Stack and Subroutines, Counter and Time Delays.

Text/Reference books:

1. Microprocessor architecture, programming and applications with 8085/8085A, Wiley eastern Ltd, by Ramesh S. Gaonkar.
2. Intel Corp: The 8085 / 8085A. Microprocessor Book – Intel marketing communication, Wiley inter science publications.
3. An introduction to micro computers Vol. 2 – some real Microprocessor – Galgotia Book Source, New Delhi by Adam Osborne and J. Kane.
4. Advanced Microprocessors by Ray and Bhurchandi - TMH.
5. Intel Corp. Micro Controller Handbook – Intel Publications.
6. Microprocessors and Interfacing by Douglas V. Hall, McGraw Hill International
7. Assembly Language Programming the IBM PC by Alan R. Miller, Subex Inc.

Semester -V**DSE-1.E1: Data Communication & Networking: 60 hours**

Introduction: Components, Application

Network Hierarchy: LAN, MAN, WAN; Topology;

Network Software: Layered, Interface, Protocol.

Reference Model: OSI and TCP/IP; Functionalities of each layer, Comparison between two models.

Data and Signals (Analog and Digital): Periodic & Non-periodic signals, Bandwidth, Bit Rate, Baud Rate, Bit Length, and Composite Signal.

Transmission Media: Transmission Spectrum, Guided (Twisted Pair, Coaxial, Optical Fiber) and Unguided (Radio Wave, Microwave, Infrared).

Transmission Impairments: Noise, Distortion and Attenuation.

Digital Transmission: Line Coding (NRZ, NRZ-L, NRZ-I, RZ, Manchester, Differential Manchester); Code Modulation (PCM, DM).

Analog Transmission: Shift Keying (ASK, FSK, PSK);

Multiplexing: FDM, TDM, WDM.

Internet: Bridges, Routers, Modem, Connectivity concept, DNS, URL, WWW, IP Addresses.

Text/ Reference Books:

1. Data Communication and Networking, B.A. Forouzan, TMH.
2. Data and Computer Communication, W. Stallings, Pearson Education.
3. Computer Network, Tanenbaum, Pearson Education.

DSE-1:E2: System Analysis & Design: 60Hours

Introduction to Software Life Cycle:

Classical and Iterative Waterfall Model; Spiral Model and its importance towards application for different system representations.

Software Requirement and Specification Analysis: Requirements Principles and its analysis principles; Specification Principles and its representations

Software Design Analysis – Different level of DFD Design, Coupling and Cohesion of the different modules Software Cost Estimation Modeling – Heuristic and Empirical Modelling; COCOMO etc.

Software Testing: Software Verification and Validation; Testing objectives, Testing Principles, Testability; Error and Faults; Unit testing and Integration testing, White Box and Black Box Testing, Test Case Design: Test Vector,

Text/ Reference Books:

1. Software Engineering: A Practitioner's Approach by R.S. Pressman, McGraw-Hill.
2. An Integrated Approach to Software Engineering by P. Jalote, Narosa Publishing House.
3. Software Engineering by K.K. Aggarwal and Y. Singh, New Age International Publishers.
4. Software Engineering by I. Sommerville, Addison Wesley.
5. Software Engineering for Students by D. Bell, Addison-Wesley.
6. Fundamentals of Software Engineering by R. Mall, PHI.

SEC-3: Introduction to Python Programming: 60 Hours

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Flowcharting, algorithms

Overview of Programming: Structure of a Python Program, Elements of Python

Introduction to Python: Python Interpreter, Using Python as calculator, Python shell, Indentation, Atoms, Identifiers and keywords, Literals, Strings, Operators(Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator)

Creating Python Programs: Input and Output Statements, Control statements (Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments, Exception handling.

Iteration and Recursion: Conditional execution, Alternative execution, Nested conditionals, Return statement, Recursion, Stack diagrams for recursive functions, Multiple assignment, while statement, for statement.

Strings and Lists: String as a compound data type, Length, Traversal and the for loop, String slices, String comparison, A find function, Looping and counting, List values, Accessing elements, List length, List ,membership, Lists and for loops, List operations, List deletion. Nested lists

Text/ Reference Books:

1. Introduction to Computation and Programming Using Python, John V. Guttag, MIT Press.
2. Think Python: How to Think Like a Computer Scientist, Allen Downey, O'Reilly.
3. Learning Python, Mark Lutz, O'Reilly.
4. Python Programming for the Absolute Beginner, Michael Dawson, Cengage Learning.
5. Learning to Program in Python, P. M. Heathcote, PG Online Limited.
6. Python Programming Fundamentals, Authors: Lee and Kent D.

Semester VI

DSE-2 E1.(a)Operating System: 60 hours

Introduction Basic OS functions, types of operating systems: batch systems–multiprogramming systems, time sharing systems;

Operating System Organization: Processor and user modes, kernels, system calls and system programs.

Process Scheduling: Scheduling criteria, Pre-emptive and non-preemptive scheduling, Long term, short term and medium term, FCFS, SJF, SRTF, Priority scheduling, Round Robin.

Process Synchronization: Concurrent Processes, critical section, semaphores and application, methods for inter-process communication;

Deadlock: Definition, Prevention, Avoidance, Detection, Recovery, Banker's algorithm.

Memory Management: Physical and virtual address space; memory allocation strategies –fixed and variable partitions, paging, segmentation, virtual memory.

DSE-2 E1. b) Operating System Lab: 40 hours

Some sample examples/Commands are given below. More problems can be included related to the theory. Use open source system (Debian OS).

1. Usage of following commands: ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd.
2. Usage of following commands: cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands: chmod, grep, tput (clear, highlight), bc.
4. Write a shell script to check if the number entered at the command line is prime or not.
5. Write a shell script to modify “cal” command to display calendars of the specified months.
6. Write a shell script to modify “cal” command to display calendars of the specified range of months.
7. Write a shell script to accept a login name. If not a valid login name display message – “Entered login name is invalid”.
8. Write a shell script to display date in the mm/dd/yy format.
9. Write a shell script to display on the screen sorted output of “who” command along with the total number of users .
10. Write a shell script to display the multiplication table any number,
11. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
12. Write a shell script to find the sum of digits of a given number.
13. Write a shell script to merge the contents of three files, sort the contents and then display them

page by page.

14. Write a shell script to find the LCD(least common divisor) of two numbers.
15. Write a shell script to perform the tasks of basic calculator.
16. Write a shell script to find the power of a given number.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.
20. Program to show the pyramid of special character “*”.

Text/ Reference Books:

1. Operating Systems Concepts, A Silberschatz, P.B. Galvin, G. Gagne, John Wiley Publications.
2. Modern Operating Systems, A.S. Tanenbaum, Pearson Education.
3. Operating Systems: A Modern Perspective, G. Nutt, Pearson Education.
4. Operating Systems, Internals & Design Principles W.Stallings, PHI.
5. Operating Systems- Concepts and design, M. Milenkovic, Tata McGraw Hill.
6. UNIX Concepts and Applications, Sumitabha Das, Tata McGraw-Hill.
7. Understanding the Linux Kernel, D. P. Bovet and M. Cesati, O'Reilly.

DSE-2. E2. a) Object Oriented Programming using C++ : 60 hours

Introduction to C and C++: Overview of Procedural Programming and Object-Orientation Programming, Compiling and Executing Simple Programs in C++. Concepts of Data Types, Variables, Constants, Operators and Basic I/O Expressions, Conditional Statements and Iterative Statements, Functions and Arrays Pointers and References in C++, Memory Allocation in C++.

Using Classes in C++: Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Class Variables & Functions, Objects as parameters, specifying the Protected and Private Access, Copy Constructors, Overview of Template classes and their use.

Overview of Function Overloading and Operator Overloading: Need of Overloading functions and operators, Overloading functions by number and type of arguments, Looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators)

Inheritance, Polymorphism and Exception Handling: Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Abstract class, Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions.

Some sample examples are given below. More problems can be included related to the theory. Use open source C++ compiler (GNU C++).

1. WAP to print the sum and product of digits of an integer.
2. WAP to reverse a number.
3. WAP to compute the sum of the first n terms of the following series $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
4. WAP to compute the sum of the first n terms of the following series $S = 1 - 2 + 3 - 4 + 5 - \dots$
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. WAP to compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.

9. WAP to print a triangle of stars as follows (take number of lines from user):

```
*  
***  
*****  
*****  
*****
```

10. WAP to perform following actions on an array entered by the user:

- i. Print the even-valued elements
- ii. Print the odd-valued elements
- iii. Calculate and print the sum and average of the elements of array
- iv. Print the maximum and minimum element of array
- v. Remove the duplicates from the array
- vi. Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

11. WAP that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments.

12. Write a program that swaps two numbers using pointers.

13. Write a program in which a function is passed address of two variables and then alter its contents.

14. Write a program which takes the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.

15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using new operator.

16. Write a menu driven program to perform following operations on strings:

- a) Show address of each character in string
- b) Concatenate two strings without using strcat function.
- c) Concatenate two strings using strcat function.
- d) Compare two strings
- e) Calculate length of the string (use pointers)
- f) Convert all lowercase characters to uppercase
- g) Convert all uppercase characters to lowercase
- h) Calculate number of vowels
- i) Reverse the string

17. Given two ordered arrays of integers, write a program to merge the two-arrays to get an ordered array.

18. WAP to display Fibonacci series (i) using recursion, (ii) using iteration

19. WAP to calculate Factorial of a number (i) using recursion, (ii) using iteration

20. WAP to calculate GCD of two numbers (i) with recursion (ii) without recursion.

21. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):

- a) Sum b) Difference c) Product d) Transpose

22. Create the Person class. Create some objects of this class (by taking information from the user). Inherit the class Person to create two classes Teacher and Student class. Maintain the respective information in the classes and create, display and delete objects of these two classes (Use Runtime

Polymorphism).

23. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.

24. Create a class Box containing length, breath and height. Include following methods in it:

a) Calculate surface Area

b) Calculate Volume

c) Increment, Overload ++ operator (both prefix & postfix)

d) Decrement, Overload -- operator (both prefix & postfix)

e) Overload operator == (to check equality of two boxes), as a friend function

f) Overload Assignment operator

g) Check if it is a Cube or cuboid

Write a program which takes input from the user for length, breath and height to test the above class.

25. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.

26. Write a program to retrieve the student information from file created in previous question and print it in following format:

Roll No. Name Marks

27. Copy the contents of one text file to another file, after removing all whitespaces.

28. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.

29. Write a program that will read 10 integers from user and store them in an array. Implement array using pointers. The program will print the array elements in ascending and descending order.

Text/ Reference Books:

1. C++: The Complete Reference, Herbert Schildt, McGraw Hill.
2. The C++ Programming Language, Bjarne Stroustrup, Addison-Wesley.
3. Object Oriented Programming with C++, E Balaguruswamy, Tata McGraw-Hill.
4. C++ How to Program, Paul Deitel, Harvey Deitel, Prentice Hall.
5. Programming with C++, John R. Hubbard, Schaum's Series.
6. Accelerated C++, Andrew Koeni, Barbara, E. Moo, Addison-Wesley.

SEC-4: Web Design: 60 hours

Introduction to HTML & CSS: Introduction to HTML, What is HTML and Where did it come from?, HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, Introduction to CSS, What is CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling. HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Advanced CSS: Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.

Introduction to JavaScript: Client-Side Scripting, What is JavaScript and What can it do?, JavaScript Design Principles, Where does JavaScript Go?, Syntax, JavaScript Objects, The Document Object Model (DOM), JavaScript Events.

Text/ Reference Books:

1. Professional JavaScript for Web Developers, Nicholas C Zakas, Wrox/Wiley India.

2. JavaScript & jQuery: The Missing Manual, David Sawyer Mcfarland, O'Reilly/Shroff Publishers & Distributors Pvt Ltd.
3. Murach's HTML5 and CSS3, Zak Ruvalcaba Anne Boehm, Murachs/Shroff Publishers & Distributors Pvt Ltd.