Punyashlok Ahilyadevi Holkar Solapur University
Faculty of Science Choice Based Credit System (CBCS),
Structure for B. Sc. (Entire Computer Science) – I
(w. e. f. 2019-20)

Subject/ Core Course	Name and Type of the Paper		No. of papers/ Practical	Hrs/week			Total Marks Per Paper	UA	CA	Credits
	Type	Name		L	T	P				
Class	B. Sc. - I Semester – I									
Ability Enhancement Course	English Paper I Part-A (communication skill)			4. 0			100	80	20	4
Core Subjects	Fundamentals of Programming using C and C++ – I	C1		2. 5	--	--	50	40	10	2
	Fundamentals of Programming using C and C++ – II			2. 5	--	--	50	40	10	2
	Computer System Architecture – I	C2		2. 5	--	--	50	40	10	2
	Computer System Architecture – II			2. 5	--	--	50	40	10	2
	Fundamental of Computer system – I	C3		2. 5	--	--	50	40	10	2
	Fundamental of Computer system – II			2. 5	--	--	50	40	10	2
	Numerical Methods – I	GE1		2. 5	--	--	50	40	10	2
	Numerical Methods – II			2. 5	--	--	50	40	10	2
Total				24	--	--	500	400	100	20
Class	B. Sc. - I Semester – II									
Ability Enhancement Course	English Paper I Part-B (communication skill)			4. 0			100	80	20	4
Core Subjects	Programming in JAVA – I	C4		2. 5	--	--	50	40	10	2
	Programming in JAVA – II			2. 5	--	--	50	40	10	2
	Discrete Structures – I	C5		2. 5	--	--	50	40	10	2
	Discrete Structures – II			2. 5	--	--	50	40	10	2
	Introduction to Web Designing – I	C6		2. 5	--	--	50	40	10	2
	Introduction to Web Designing – II			2. 5	--	--	50	40	10	2
	Digital Electronics – I	GE-2		2. 5	--	--	50	40	10	2
	Digital Electronics – II			2. 5	--	--	50	40	10	2
Total (Theory)				24	--	--	500	400	100	20
Core Subjects	C1 & C3		Pract. I	--	--	4	100	80	20	4. 0
	C4		Pract. II	--	--	4	100	80	20	4. 0
	C6		Pract III	--	--	4	100	80	20	4. 0
	C5 &GE1		Pract IV	--	--	4	100	80	20	4. 0
	C2 & GE2		Pract. V	--	--	4	100	80	20	4. 0
Total (Practical)					20		500	400	100	20
Grand Total				48		20	1500	1200	300	60

CORE COURSES (ENTIRE COMPUTER SCIENCE)

English Paper I Part-A (communication skill)

C-1: Fundamentals of Programming using C and C++

Theory: 60 Lectures

Section – I

1. **Introduction to C and C++** **(3 Lectures)**
   History of C and C++, Overview of Procedural Programming and Object-Orientation Programming, Using main() function, Compiling and Executing Simple Programs in C++.
2. **Data Types, Variables, Constants, Operators and Basic I/O** **(5 Lectures)**
   Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putcharetc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio. h, iostream. h, conio. hetc).
3. **Expressions, Conditional Statements and Iterative Statements** **(5 Lectures)**
   Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)
4. **Functions and Arrays** **(10 Lectures)**
   Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments. Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, accessing individual elements in an Array, manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two-Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays
5. **Derived Data Types (Structures and Unions)** **(3 Lectures)**
   Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.
6. **Pointers and References in C++** **(7 Lectures)**
   Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with 11 Pointers, passing pointers as function arguments, returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values

Section – II

7. **Memory Allocation in C++** **(3 Lectures)**
   Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, use of new and delete operators, storage of variables in static and dynamic memory allocation.

8. **File I/O, Preprocessor Directives** **(4 Lectures)**
   Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros.

9. **Using Classes in C++** **(7 Lectures)**
   Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables &Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors, Overview of Template classes and their use.

10. **Overview of Function Overloading and Operator Overloading** **(5 Lectures)**
    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)

11. **Inheritance, Polymorphism and Exception Handling** **(8 Lectures)**
    Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.

Reference Books:
1. Herbtz Schildt, C++: The Complete Reference, Fourth Edition, McGraw Hill. 2003
2. Bjarne Stroustrup, The C++ Programming Language, 4th Edition, Addison-Wesley, 2013.
3. Bjarne Stroustroup, Programming -- Principles and Practice using C++, 2nd Edition, Addison-Wesley 2014.
4. E Balaguruswamy, Object Oriented Programming with C++, Tata McGraw-Hill Education, 2008.
5. Paul Deitel, Harvey Deitel, C++ How to Program, 8th Edition, Prentice Hall, 2011.
6. John R. Hubbard, Programming with C++, Schaum's Series, 2nd Edition, 2000.
7. Andrew Koeni, Barbara, E. Moo, Accelerated C++, Published by Addison-Wesley, 2000.
8. Scott Meyers, Effective C++, 3rd Edition, Published by Addison-Wesley, 2005.
9. Harry, H. Chaudhary, Head First C++ Programming: The Definitive Beginner's Guide, First Create space Inc, O-D Publishing, LLC USA 2014.
10. Walter Savitch, Problem Solving with C++, Pearson Education, 2007.
11. Stanley B. Lippman, Josee Lajoie, Barbara E. Moo, C++ Primer, Published by Addison-Wesley, 5th Edition, 2012.

C-2: Computer System Architecture

Theory: 60 Lectures

Section – I

1. **Introduction** **(8 lectures)**
   Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.
2. **Data Representation and Basic Computer Arithmetic** **(10 lectures)**
   Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers
3. **Basic Computer Organization and Design** **(13 lectures)**
   Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.

Section – II

4. **Central Processing Unit** **(15 lectures)**
   Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming, RISC, CISC architectures, pipelining and parallel architecture.
5. **Memory Organization** **(6 lectures)**
   Cache memory, Associative memory, mapping.
6. **Input-Output Organization** **(8 lectures)**
   Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

Reference Books:
1. M. Mano, Computer System Architecture, Pearson Education 1992
2. A. J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology, 2004
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India,2009
4. M. M. Mano , Digital Design, Pearson Education Asia,2013
5. Carl Hamacher, Computer Organization, Fifth edition, McGrawHill, 2012.

C-3: Fundamental of Computer system

Theory: 60 Lectures

Section – I

1. **Introduction to computer** (6 Lectures)
   Definition of System and computer, characteristics, limitations, concepts of h/w and s/w, capabilities of computers, evaluation, generation, classification based on size and purpose.
2. **Introduction to Information Technology** (10 Lectures)
   Information Technology: Definition, Introduction, Information Systems, Data and information; IT in Business, Industry, Home, Play, Education, Training, Science and Engineering.
3. **Computer languages** (14 Lectures)
   Low level; Assembly level, High level; types of Languages: imperative, declarative, rule based, object oriented; tools in software development kit: translators (Compiler, interpreter, Assembler, Linker, Loader and debuggers), Editors, Header files, namespaces, packages, Libraries, Virtual Machines (CLR, JVM), IDE.

Section – II

4. **Structure of modern computer** (10 Lectures)
   Block diagram, ALU, Memory Unit, Control Unit, Introduction to motherboard, SMPS, Expansion Slots, Serial and Parallel ports
5. **Input and Output Devices** (10 Lectures)
   Input devices: - Character input device, Pointing input device, Audio input device, Video/Image input Device, sensors. Output devices: - Hard copy and soft copy output devices, types of printers and Monitors, Virtual and Augmented reality based input output devices.
6. **Concept of Memory** (10 Lectures)
   Primary Memory (RAM and Various types of ROMS) and Secondary Storage devices, types of Memory based on materials: electronic, optical and magnetic; RAID and its levels.

Reference Books:

1. Computer Today --Basandra
2. Fundamental of computers --V. Raja Raman.
3. Computer Fundaments --P. K. Sinha.
4. IBM PC and clones by Govinda Rajulu

GE-1: Numerical Methods

Theory: 60 Lectures

Section – I

1. **Computer Arithmetic and Errors** **(8 Lectures)**
   Floating point representation and computer arithmetic, Significant digits, Errors: Round-off error, Local truncation error, Global truncation error, absolute error, relative error and percentage error, Convergence and terminal conditions.
2. **Solutions of Nonlinear Equations** **(10 Lectures)**
   Location of roots, Bisection method, Regula−Falsi method and it's rate of convergence, Newton−Raphson method and it's rate of convergence, comparison of these methods.
3. **Solution of System of linear Equations** **(12 Lectures)**
   Matrices, elementary matrices, System of linear equations, solution of system by using Gauss elimination method (with row pivoting) and Gauss−Jordan elimination method, Iterative methods: Jacobi and Gauss-Seidel iterative methods.

Section – II

4. **Interpolation:** **(10 Lectures)**
   Finite difference operators, relation between these operators, Gregory Newton forward and backward differences Interpolation, Lagrange's interpolation formula, Linear interpolation: Newton's divided difference formula (only formula without proof), problems.
5. **Numerical differentiation and Integration** **(10 Lectures)**
   Numerical values of First order derivatives and second order derivatives, Numerical integration: General quadrature formula for equidistant ordinates, Trapezoidal rule, Simpson's 1/3rd & 3/8th rule, Newton−Cotes open formulas without proof,
6. **Extrapolation Ordinary differential equation** **(10 Lectures)**
   Degree and order of a differential equation, Definition of ordinary differential equation, Taylor's series method, Euler's method, Modified Euler's methods, Runge-Kutta second order method, Classical 4th order Runge-Kutta method

Reference Books:
1. Balguruswamy, Computers and Numerical Methods (TMH).
2. M. K. Jain, S. R. K. Iyengar and R. K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age International Publisher, 6/e (2012).
3. Steven C Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists, Tata McGraw Hill, 2/e (2010).
4. Introduction to Numerical Analysis by S. S. Sastri, Tata McGraw Hill.

English Paper I Part-B (communication skill)

C4: Programming in JAVA

Theory: 60 Lectures

Section – I

1. **Introduction to Java** **(12 Lectures)**
   Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods).

2. **Arrays, Strings and I/O** **(10 Lectures)**
   Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System. out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

3. **Object-Oriented Programming Overview** **(8 Lectures)**
   Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

   Section – II

5. **Inheritance, Interfaces, Packages, Enumerations** **(12 lectures)**
   Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

6. **Exception Handling, Networking and Database Connectivity** **(10 Lectures)**
   Exception types, uncaught exceptions, throw, built-in exceptions, creating your own exceptions; Multi-threading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java. net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

7. **Applets and Event Handling** **(8 Lectures)**
   Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as Labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

Reference Books:

1. Ken Arnold, James Gosling, David Homes, The Java Programming Language, 4th Edition, 2005.
2. James Gosling, Bill Joy, Guy L Steele Jr, Gilad Bracha, Alex Buckley The Java Language Specification, Java SE 8 Edition (Java Series), Published by Addison Wesley, 2014.
3. Joshua Bloch, Effective Java 2nd Edition, Publisher: Addison-Wesley, 2008.
4. Cay S. Horstmann, Gary Cornell, Core Java 2 Volume 1, 9th Edition, Printice Hall. 2012
5. Cay S. Horstmann, Gary Cornell, Core Java 2 Volume 2 - Advanced Features, 9th Edition, Printice Hall. 2013.
6. Bruce Eckel, Thinking in Java, 3rd Edition, PHI, 2002.
7. E. Balaguruswamy, Programming with Java, 4th Edition, McGraw Hill. 2009.
8. Paul Deitel, Harvey Deitel, Java: How to Program, 10th Edition, Prentice Hall, 2011.
9. David J. Eck, Introduction to Programming Using Java, Published by CreateSpace Independent Publishing Platform, 2009.
10. John R. Hubbard, Programming with Java, Schaum's Series, 2nd Edition, 2004.

C5: Discrete Structures

Theory 60 Lectures

Section – I

1. **Introduction to sets and Relation** **(12 Lectures)**
   Sets - finite and Infinite sets, uncountably Infinite Sets; relations, types of relation: identity, reflexive, symmetric, equivalence, antisymmetric, partial ordering, asymmetric, Closure, transitive closure, Warshall's algorithm, Partial counting - Pigeonhole Principle, Principle of Inclusion and Exclusion, Mathematical Induction.
2. **Functions:** **(8 Lectures)**
   Functions- Definition of function as relation, domain, co-domain and range of a function, injective, surjective and bijective functions, inverse function, composition of functions.
3. **Recurrence Relation:** **(10 Lectures)**
   Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, homogeneous Recurrence Relations with constant coefficients and their solution, total solution.

Section – II

4. **Graph Theory** **(10 Lectures)**
   Basic Terminology, Types of graphs: simple, multi, pseudo, null, regular, r-regular, complete, bipartite, complete bipartite, weighted graphs, Degree of a vertex, total degree of a graph, shaking hand lemma and elementary results, Matrix Representation of graphs: Adjacency and incidence matrix, Isomorphism of graphs.
5. **Derived Graphs** **(8 Lectures)**
   Sub graph, vertex deleted, and edge deleted subgraphs, vertex disjoint and edge disjoint subgraphs, Operations on Graphs: Union, intersection and ring sum of two graphs, Product of two graphs.
6. **Euler and Hamiltonian Graph** **(12 Lectures)**
   Walk, trail, path, Euler trail and circuit, Euler's graph, Hamiltonian Path and Circuit, Hamiltonian Graph, travelling sales man problem, Chinese Postman problem Trees: Basic Terminology and properties of Trees, Spanning Trees, Shortest spanning tree, Kruskal's algorithm for shortest spanning tree, branches & chords, fundamental cut sets & circuits

Recommended Books:
1. C. L. Liu, D. P. Mahopatra, Elements of Discrete mathematics, 2nd Edition, Tata McGraw Hill, 1985.
2. Kenneth Rosen, Discrete Mathematics and Its Applications, 6th Edition, McGraw Hill 2006.
3. T. H. Coremen, C. E. Leiserson, R. L. Rivest, Introduction to algorithms, 3rd edition Prentice Hall on India, 2009.
4. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms, John Wiley Publication, 1988.
5. J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009.
6. D. J. Hunter, Essentials of Discrete Mathematics, Jones and Bartlett Publishers, 2008

C-6: Introduction to Web Designing

Theory 60 Lectures

Section – I

1. **Web Design Principles**   **(5 Lectures)**
   Basic principles involved in developing a web site, Planning process, Five Golden rules of web designing, Designing navigation bar, Page design, Home Page Layout, Design Concept.

2. **Basics in Web Design**   **(5 Lectures)**
   Brief History of Internet, what is World Wide Web, Why create a web site, Web Standards, Audience requirement.

3. **Introduction to HTML**   **(8 Lectures)**
   What is HTML, HTML Documents, Basic structure of an HTML document, Creating an HTML document, Mark up Tags, Heading-Paragraphs, Line Breaks, HTML Tags.

Section – II

4. **Elements of HTML**   **(12 Lectures)**
   Introduction to elements of HTML, Working with Text, Working with Lists, Tables and Frames, Working with Hyperlinks, Images and Multimedia, Working with Forms and controls.

5. **Introduction to Cascading Style Sheets**   **(18 Lectures)**
   Concept of CSS, Creating Style Sheet, CSS Properties, CSS Styling(Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model(Introduction, Border properties, Padding Properties, Margin properties), CSS Advanced(Grouping, Dimension, Display, Positioning, Floating, Align, Pseudo class, Navigation Bar, Image Sprites, Attribute sector), CSS Colour, Creating page Layout and Site Designs.

6. **Introduction to Web Publishing or Hosting**   **(12 Lectures)**
   Creating the Web Site, Saving the site, working on the web site, creating web site structure, Creating Titles for web pages, Themes-Publishing web sites.

Reference Books:

1. Web Designing & Architecture-Educational, Technology Centre University of Buffalo
2. Steven M. Schafer, HTML, XHTML, and CSS Bible, 5ed Wiley India
3. John Duckett Beginning HTML, XHTML, CSS, and JavaScript, Wiley India
4. Ian Pouncey, Richard York Beginning CSS: Cascading Style Sheets for Web Design, Wiley India
5. Kogent Learning Web Technologies: HTML, Javascript Wiley India

GE-2: Digital Electronics

Theory 60 Lectures

Section – I

1. **Number Systems and Arithmetic**   **(08 Lectures)**
   Decimal Number System & Binary Number System Decimal to Binary conversion (Double-dabble method only), Binary to Decimal Conversion, excess-3 code, gray code, Binary Arithmetic: Binary addition, subtraction, multiplication & division Hexadecimal number system, Hexadecimal to binary, binary to Hexadecimal, Hexadecimal to decimal conversion Hexadecimal arithmetic: Addition, subtraction, multiplication & division Binary subtraction using 1' complement, 2's complement.
   method.

2. **Logic Gates**   **(10 Lectures)**
   Boolean Algebra and Logic Gates:- Logic Gates : AND, OR, NOT, Ex-OR, Ex-NOR NAND as Universal building block, Logic diagrams of Boolean expressions Postulates of Boolean Algebra Theorems of Boolean Algebra, De Morgan's theorems Reducing Boolean expressions Minimization Techniques Introduction, Minterms and Maxterms, K-Map, K-map for 2 variables K-map for 3 variables K-map for 4 variables, error detection & correction tech. (parity check, hamming code)

3. **Combinational and Arithmetic Logic Circuits**   **(10 Lectures)**
   Half Adder & Full Adder, Binary parallel Adder, Half Subtractor, Full Subtractor, universal Adder/Subtractor, Multiplexer and Demultiplexer, different types of Multiplexer and Demultiplexer, encoder and decoder, pin function of IC 74150, 74154, 74138, 74148.

Section – II

4. **Flip Flops**   **(10 Lectures)**
   Concept of flip-flop, types of F/F, RS F/F, Clocked RS F/F, D F/F Triggering (Positive, negative), preset and clear F/F, JK F/F, T F/F, Race around condition of JK F/F, Master slave JK F/F.

5. **Counters**   **(10 Lectures)**
   Introduction to counter, types of counters-synchronous, Asynchronous, Synchronous counter : 3-bit up counter, down counter, up-down counter, Synchronous serial & synchronous parallel counter, BCD counter, Ring counter, Johnson counter, pin configuration of IC 7490 Asynchronous counter / ripple counter: Modulus Counter, MOD-2, 5, 10 counter Shift Registers Introduction register, types of shift register: Serial- in serial –out(left shift register, right shift register, combined left & right), Serial-in parallel-out**,** Parallel-in serial-out, parallel-in parallel-out, pin configuration of IC 7495, shift counter, Ring counter.

6. **Data Converters**   **(10 Lectures)**
   Basic concepts of DAC and ADC, specifications Digital to analog conversion: Binary weighted and R - 2 R ladder networks Analog to digital conversion: Successive approximation method, Dual slope Semiconductor Memories**:** Memory cell, Memory organization, operation and parameters. types of memory, RAM (Static, Dynamic), pin connection of RAM chip, Classification of ROM (PROM, EPROM), dot matrix PROM, read-write operation of memory, memory parameter, Flash memory.

Reference Books
1. Digital principle & applications – Malvino Leech.
2. Digital principle – Floyed.
3. Digital electronics – C. F. Strangio.
4. Modern Digital electronics – R. P. Jain.
5. Principle of electronics – B. V. Narayan Rao (3rd edition).

LAB (C-1): Programming Fundamentals using C/C++

Practical: 60 Lectures

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+ ……
4. WAP to compute the sum of the first N terms of the following series S =1-2+3-4+5…………….
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() or 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): 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.

LAB (C-2): Computer System Architecture

Lab Practical: 60 Lectures

1. Create the micro operations and associate with instructions as given in the chapter
2. Create a Fetch routine of the instruction cycle.
3. Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

a. CLA	e. CIR	i. SNA
b. CLE	f. CIL	j. SZA
c. CMA	g. INC	k. SZE
d. CME	h. SPA	l. HLT

4. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.
    a. ADD f. BSA
    b. AND g. ISZ
    c. LDA
    d. STA
    e. BUN
5. Simulate the machine for the memory-reference instructions referred in above question with I= 1 and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.
6. Modify the machine created in Practical 1 according to the following instruction format:
    a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. There are only two addressing modes, I = 0 (direct addressing) and I = 1 (indirect addressing).
    b. Create a new register I of 1 bit.
    c. Create two new microinstructions as follows:
        i.   Check the opcode of instruction to determine type of instruction (Memory Reference/Register Reference/Input-Output) and then jump accordingly.
        ii. Check the I bit to determine the addressing mode and then jump accordingly.

Lab C-3 Fundamental of Computer System

Practical: 60 Lectures

1. Practice on Text editing tools: notepad, word pad, open org, MS-word, Text pad etc.
2. Practice on Spreadsheet tools: spreadsheet, numbers, Libreoffice calc, open office calc, google sheet etc.
3. Presentation tools: PowerPoint, open office slideshow, Keynote etc.
4. Windows powerShell Commands: Internal and external, net command batch files.

LAB (C-4): Programming in Java

Practical: 60 Lectures

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of length in case of a two-dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBufferclasss like setCharAt(), setLength(), append(), insert(), concat()and equals().
9. Write a program to create a —distance‖ class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer
10. Modify the —distance‖ class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.
11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions (from lower to higher data type)
12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword
13. Write a program to show the use of static functions and to pass variable length arguments in a function.
14. Write a program to demonstrate the concept of boxing and unboxing.
15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).
16. Write a program to create a multilevel package and creates a reusable class to generate Fibonacci series, where the function to generate Fibonacci series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program —Divide by Zero that takes two numbers a and b as input, computes a/b, and invokes Arithmetic Exception to generate a message when the denominator is zero.
19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).
21. Write a program to demonstrate priorities among multiple threads.
22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
23. Write a program to create URL object, create a URLConnection using the openConnection() method and then use it examine the different components of the URLand content.

24. Write a program to implement a simple datagram client and server in which a message that is typed into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet's window.
26. Write a program to get the URL/location of code (i. e. java code) and document (i. e. html file).
27. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
28. Write a program to demonstrate different keyboard handling events.
29. Write a program to generate a window without an applet window using main() function.
30. Write a program to demonstrate the use of push buttons.

Lab C5: Discrete Structures

Practical: 60 Lectures

1. Relations- Diagraph of relations, transitive closure by using Warshall's algorithm.
2. Matrix representation of graph: Adjacency and incidence matrix.
3. Operations on graphs: Union, intersection, ring sum, product of two graphs.
4. Solution of Chinese Postman Problem for both Euler's and non-Euler's Graph.
5. Solution of Travelling Salesman Problem
6. Kruskal's algorithm for weighted spanning tree
7. Fundamental circuits and fundamental cut sets

Lab C6-Introduction to web designing

1. Design a page having suitable background colour and text colour with title "My First Web Page" using all the attributes of the Font tag.
2. Create a HTML document giving details of your [Name, Age], [Address, Phone] and [Register Number, Class] aligned in proper order using alignment attributes of Paragraph tag.
3. Write HTML code to design a page containing some text in a paragraph by giving suitable heading style.
4. Create a page to show different character formatting (B, I, U, SUB, SUP)
5. Write HTML code to create a Web Page that contains an Image at its center.
6. Create a web page with an appropriate image towards the left-hand side of the page, when user clicks on the image another web page should open.
7. Create web Pages using Anchor tag with its attributes for external links.
8. Create a web page for internal links; when the user clicks on different links on the web page it should go to the appropriate locations/sections in the same page.
9. Write a HTML code to create a web page with pink colour background and display moving message in red colour.
10. Create a web page, showing an ordered list of all second semester courses (Subjects).
11. Create a web page, showing an unordered list of names of all the Diploma Programmes (Branches) in your institution.
12. Create a HTML document containing a nested list showing a content page of any book.
13. Create the table in HTML with Dummy Data

Lab GE-1: Numerical Methods practical

Practical: 60 Lectures

1. Find the solution of system of equations by using Gauss method and Gauss Jordan Method.
2. Find the solution of system of equations using Jacobi and Gauss-Seidel method.
3. Find the roots of the equation by bisection method.
4. Find the roots of the equation by Regula−Falsi method.
5. Interpolation: Newton's both forward and backward interpolation, Lagrange's interpolation.
6. Numerical integration: Trapezoidal rule, Simpson's 1/3rd and 3/8th rule.
7. Find the roots of the equation by bisection method.
8. Find the roots of the equation by secant/Regula−Falsi method.
9. Find the roots of the equation by Newton's method.
10. Find the solution of a system of nonlinear equation using Newton's method.
11. Find the solution of tridiagonal system using Gauss Thomas method.
12. Find the cubic spline interpolating function.
13. Evaluate the approximate value of finite integrals using Gaussian/Romberg integration.
14. Solve the boundary value problem using finite difference method.

Note: Programming is to be done in any one of Computer Algebra Systems: MATLAB / MATHEMATICA / MAPLE/ C / C++.

Lab GE-2: Digital Electronics Practical

Practical: 60 Lectures

1. Study of basic gates
2. Study of Flip-flop (RS, D)
3. Study of binary counter
4. Study of DE Morgan's theorems
5. Adder using gates
6. Interco version of gates using Universal Gates (NOR/NAND)
7. Assembly Language program for Arithmetic operations using 8085 Microprocessor
8. Assembly Language program for Ascending and descending numbers using 8085 Microprocessor
9. Assembly Language program for addressing modes in 8085 Microprocessor
10. Assembly Language program for Block Transfer using 8085 Microprocessor
11. Assembly Language program for Stack using 8085 Microprocessor
12. Addressing modes of 8086
13. Arithmetic operations of 8086
14. Arrange ascending and descending of given data
15. Interfacing card 8255 (Mode 0)
16. Thumb wheal interfacing using 8255
17. Seven segment interfacing 8253 (Mode 2)
18. Study of single board computer and stack related operations
19. Keyboard interfacing using 8279

Note: - Assembly Language program in 8085 and 8086 must be implemented using 8085 and 8086 Microprocessor Kit.