

Revised
SYLLABUS FOR
Bachelor of Science (Honours)

COMPUTER SCIENCE

THREE YEAR DEGREE COURSE
SEMESTER SYSTEM

(Under New UGC CBCS Guidelines)

2021

COURSE STRUCTURE

SEMESTER	COURSE	COURSE NAME	COURSE CODE	CREDIT
I	Core 1	Programming Fundamentals using C(Theory)	CSC 1.11	4
		Programming Fundamentals using C (Practical)	CSC 1.12	2
	Core2	Computer System Architecture(Theory)	CSC 1.21	4
		Computer System Architecture(Practical)	CSC 1.22	2
II	Core 3	Data Structures(Theory)	CSC 2.11	4
		Data Structures(Practical)	CSC 2.12	2
	Core 4	Programming in Java(Theory)	CSC 2.21	4
		Programming in Java(Practical)	CSC 2.22	2
III	Core 5	Object Oriented Programming in C++ (Theory)	CSC 3.11	4
		Object Oriented Programming in C++(Practical)	CSC 3.12	2
	Core 6	Operating Systems(Theory)	CSC 3.21	4
		Operating Systems(Practical)	CSC 3.22	2
	Core 7	Computer Networks(Theory)	CSC 3.31	4
		Computer Networks(Practical)	CSC 3.32	2
	Skill Enhancement Course 1	HTML	CSS 3.11	2
	IV	Core 8	Design and Analysis of Algorithms(Theory)	CSC 4.11
Design and Analysis of Algorithms(Practical)			CSC 4.12	2
Core 9		Software Engineering(Theory)	CSC 4.21	4
		Software Engineering(Practical)	CSC 4.22	2
Core 10		Database Management Systems(Theory)	CSC 4.31	4
		Database Management Systems(Practical)	CSC 4.32	2
Skill Enhancement Course 2	Programming in MATLAB	CSS 4.11	2	
V	Core 11	Internet Technologies(Theory)	CSC 5.11	4
		Internet Technologies(Practical)	CSC 5.12	2
	Core 12	Theory of Computation (Theory)	CSC 5.21	5
		Theory of Computation(Tutorial)	CSC 5.22	1
	Discipline Specific Elective 1	PHP Programming (Theory)	CSD 5.11	4
		PHP Programming(Practical)	CSD 5.11	2
	Discipline Specific Elective 2	Programming in Visual Basic(Theory)	CSD 5.21	4
		Programming in Visual Basic(Practical)	CSD 5.22	2
VI	Core 13	Artificial Intelligence(Theory)	CSC 6.11	5
		Artificial Intelligence(Tutorial)	CSC 6.12	1
	Core 14	Computer Graphics(Theory)	CSC 6.21	4
		Computer Graphics(Practical)	CSC 6.22	2
	Discipline Specific Elective 3	Cloud Computing (Theory)	CSD 6.11	4
		Cloud Computing (Practical)	CSD 6.12	2
	Discipline Specific Elective 4	Project Work / Dissertation	CSD 6.21	6

SEMESTER – I

CORE 1 (CSC 1.11) PROGRAMMING FUNDAMENTALS USING C

Theory Credit: 4

Teaching Hours: 60

- UNIT I** Overview of Procedure oriented programming, Data types, Defining and Initializing Variables, Scope of Variables, Keywords, Casting of Data Types, Operators (Arithmetic, Relational, and Logical), main() function, Compiling and Executing a program, Using Comments in programs, Character I/O (getc, getchar, putc), printf(), scanf(), Basic Header Files (stdio.h, conio.h, math.h, string.h, etc).
- UNIT II** Simple Expressions in C, Understanding precedence of Operators in Expressions, Conditional Statements (IF, IF... ELSE, Nested IF, Switch-Case), Iterative Statements (FOR, WHILE, and DO-WHILE), Use of BREAK and CONTINUE in Loops, Nested loops.
- UNIT III** Use of functions, Call by Value, Call by Reference, Functions returning value, Void function, One Dimensional Arrays (Declaring 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, Working with Rows and Columns of a matrix.
- UNIT IV** **Understanding** use of structures, declaring, initializing and using simple structures, Manipulating individual members of structures, Array of Structures
- UNIT V** Understanding Pointer Variable, Simple use of Pointers, Pointer arithmetic, Differentiating between static and dynamic memory allocation, use of malloc and calloc functions, Simple file handling operations.

Recommended Books and References:

1. Programming in ANSI C - E. Balguruswamy, TMH
2. Let us C - Yaswant Kanitkar, BPB

CORE 1 (CSC 1.12) C PROGRAMMING LAB

1. Write a program to find the maximum among three numbers.
2. Write a program to find factorial of a number.
3. Write a program to print the Fibonacci series.
4. Write a program to print the prime numbers between 1 to n.
5. Write a program to check if it a Palindrome (e.g., madam).
6. Write a program to check if it is an Armstrong number. (An Armstrong number of three digits is an integer such that the sum of the cubes of its digits is equal to the number itself)
7. Write program to reverse the digits of an input number.
8. Write a program to find the sum of digits of a number.
9. Write a program to find the GCD and LCM of two numbers.
10. Write a program to perform different arithmetic operations using switch....case.
11. Write a program to count the number bits "1" in a given binary number.
12. Write a program to find the factorial of a number using function.

13. Write a program to perform the arithmetic operations using function.
14. Write a program to find the largest element in an array.
15. Write a program to add two matrices.
16. Write a program to multiply two matrices.
17. Write a program to find the position of a given character in a string.
18. Write a program to count the number of times a character occurs in a string.
19. Write a program to create records of 10 students using structure.

CORE 2 (CSC 1.21)
COMPUTER SYSTEM ARCHITECTURE

Theory Credit: 4

Teaching Hours: 60

- UNIT I Introduction:** Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.
- UNIT II Data Representation and Basic Computer Arithmetic:** Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers
- UNIT III Basic Computer Organization and Design:** 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.
- UNIT IV Central Processing Unit:** 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.
- UNIT V Input-Output Organization:** Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

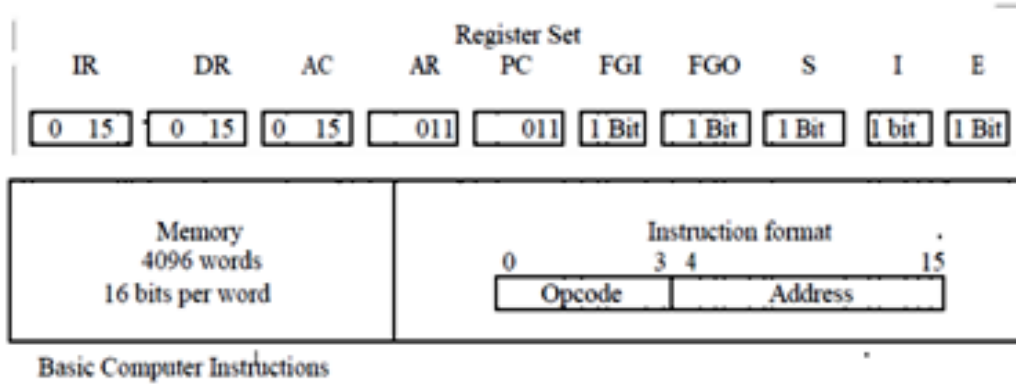
Recommended Books and References:

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, 2004th
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8 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.

CORE 1 (CSC 1.22)
COMPUTER SYSTEM ARCHITECTURE LAB

1. Create a machine based on the following architecture:

I. Create a machine based on the following architecture:



Memory Reference		Register Reference		Input-Output	
Symbol	Hex	Symbol	Hex	Symbol	Hex
AND	0xxx	CLA	E800	INP	F800
ADD	2xxx	CLE	E400	OUT	F400
LDA	4xxx	CMA	E200	SKI	F200
STA	6xxx	CME	E100	SKO	F100
BUN	8xxx	CIR	E080	ION	F080
BSA	Axxx	CIL	E040	IOF	F040
ISZ	Cxxx	INC	E020		
AND_I	1xxx	SPA	E010		
ADD_I	3xxx	SNA	E008		
LDA_I	5xxx	SZA	E004		
STA_I	7xxx	SZE	E002		
BUN_I	9xxx	HLT	E001		
BSA_I	Bxxx				
ISZ_I	Dxxx				

Refer to Chapter-5 of Morris Mano for description of instructions.

2. Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.
3. Create a Fetch routine of the instruction cycle.
4. 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
 - b. CLE
 - c. CIL
 - d. CIR
 - e. SNA
 - f. SZA
 - g. SKI
 - h. SKO
 - i. ION
 - j. IOF

Optional

- c. CMA
- d. CME
- g. INC
- h. SPA
- k. SZE
- l. HLT

5. 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
 - b. AND
 - c. LDA
 - d. STA
 - e. BUN
 - f. BSA
 - g. ISZ
6. 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.

SEMESTER – II

CORE 3 (CSC 2.11) DATA STRUCTURES

Theory Credit:

Teaching Hours: 60

- UNIT I Arrays** Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation).
- UNIT II Stacks** Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack.
- UNIT III Linked Lists:** Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists.
- UNIT IV Trees:** Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees).
- UNIT V Searching and Sorting:** Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques.

Recommended Books and References:

1. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.
2. Sartaj Sahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
3. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using C and C++", Second edition, PHI, 2009.
4. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson, 1999.
5. D.S Malik, Data Structure using C++, Second edition, Cengage Learning, 2010.
6. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
7. Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidyah Langsam, "Data Structures Using Java, 2003.
8. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub, 2003
9. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 edition, 2009
10. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley, 2013
11. Herbert Schildt, "Java The Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.
12. D. S. Malik, P.S. Nair, "Data Structures Using Java", Course Technology, 2003.

CORE 3 (CSC 2.12)
DATA STRUCTURES LAB

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. (ii) WAP to display fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree:
 - (a) Insertion (Recursive and Iterative Implementation)
 - (b) Deletion by copying
 - (c) Deletion by Merging
 - (d) Search a no. in BST
 - (e) Display its preorder, postorder and inorder traversals Recursively
 - (f) Display its preorder, postorder and inorder traversals Iteratively
 - (g) Display its level-by-level traversals
 - (h) Count the non-leaf nodes and leaf nodes
 - (i) Display height of tree
 - (j) Create a mirror image of tree
 - (k) Check whether two BSTs are equal or not
15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement various operations on AVL Tree.

CORE 4 (CSC 2.21)
PROGRAMMING IN JAVA

Theory Credit:

Teaching Hours: 60

UNIT I Introduction to Java: 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),

UNIT II Arrays, Strings and I/O: 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.

UNIT III Object-Oriented Programming Overview: 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.

Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata: 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.

UNIT IV Exception Handling, Threading, Networking and Database Connectivity: 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.

UNIT V Applets and Event Handling: 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, text fields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

Recommended Books and References:

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. "Head First Java", Orielly Media Inc. 2nd Edition, 2005.

CORE 4 (CSC 2.22)
PROGRAMMING IN JAVA LAB

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 StringBuffer classes like setCharAt(), setLength(), append(), insert(), concat() and equals().
9. Write a program to create a –distance1 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 –distance1 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 also 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 –DivideByZero1 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.

SEMESTER – III

CORE 5 (CSC 3.11)

OBJECT ORIENTED PROGRAMMING IN C++

Theory Credit:

Teaching Hours: 60

- UNIT I** Object Oriented Programming concept, Procedural vs OOP programming, OOP terminology and features, Tokens, Character set, Keywords, Data-types, Constants and variables, expressions, Standard Library and header files. Operator and Expressions: Arithmetic Operator, Increment/Decrement Operator, Relational Operator, Logical Operator and conditional operators, library functions, Logical Expressions.
- UNIT II** Control statements, IF, IF ...ELSE, Nested IF, Switch....Case, Looping statements, While, Do-while, For statements, nested loops.
Classes and Objects: Need for Classes, Declaration of Classes, referencing class Members, Data members and member Functions, Inline Functions, Creation of objects
- UNIT III** Use of access specifiers, Public and Private, Function Overloading, use of Constructors and Destructors, Types of constructors: default, parameterized, and copy constructors, Operator overloading, Friend function, Arrays of objects.
- UNIT IV** 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.
- UNIT V** Concept of Inheritance, Types of inheritance: Single level, multi-level, multiple, hybrid, Use of protected access specifier, Function overriding, Exception handling, Simple file handling.

Recommended Books and References:

1. Object Oriented Programming with C++ -E. Balaguruswamy, TMH
2. Let us C++ -Yashavant Kanetkar, BPB
3. Object Oriented Programming with C++ -Sourav Sahay, Oxford University Press
4. The C++ Programming Language - Bjarne Stroustrup, Addison-Wesley

CORE 5 (CSC 3.12)

C++ PROGRAMMING LAB

1. Define a class called STUDENT with the data members Roll No., Name, Marks secured in five subjects. Write member functions to do the following:
 - i. Read data
 - ii. Find the total mark and division
 - iii. Display Roll No., Name, Total mark, and Division
2. Define a class called SHAPE with appropriate data members. Find the area of different geometrical shapes using function overloading.

3. Define a class called ACCOUNT with the data members Account no. Customer name, Amount and initialize with suitable constructor. Write member functions to do the following:
 - i. Deposit amount
 - ii. Withdraw amount
 - iii. Check balance
4. Using operator overloading add two given Lengths expressed as Feet and Inch.
5. Using operator overloading add two given TIMEs expressed as Hour : Minute : Second.

CORE 6 (CSC 3.21) OPERATING SYSTEMS

Theory Credit: 4

Teaching Hours: 60

UNIT I Introduction: Basic OS functions, resource abstraction, types of operating systems– multiprogramming systems, batch systems, time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

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

UNIT III Process Management: System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent and processes, critical section, semaphores, methods for inter-process communication; deadlocks.

UNIT IV Memory Management: Physical and virtual address space; memory allocation strategies -fixed and variable partitions, paging, segmentation, virtual memory

UNIT V File and I/O Management: Directory structure, file operations, file allocation methods, device management.

Recommended Books and References:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles , 5th Hall of India. 2008. Edition, Prentice.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

CORE 6 (CSC 3.22) OPERATING SYSTEMS LAB

C/ C++ programs

1. WRITE A PROGRAM (using *fork()* and/or *exec()* commands) where parent and child execute:
 - a) same program, same code.
 - b) same program, different code.

- c) before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
 3. WRITE A PROGRAM to report behavior of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
 4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
 5. WRITE A PROGRAM to copy files using system calls.
 6. Write program to implement FCFS scheduling algorithm.
 7. Write program to implement Round Robin scheduling algorithm.
 8. Write program to implement SJF scheduling algorithm.
 9. Write program to implement non-preemptive priority based scheduling algorithm.
 10. Write program to implement preemptive priority based scheduling algorithm.
 11. Write program to implement SRJF scheduling algorithm.
 12. Write program to calculate sum of n numbers using *thread* library.
 13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

CORE 7 (CSC 3.31)
COMPUTER NETWORKS

Theory Credit: 4

Teaching Hours: 60

- UNIT I** Introduction to Computer Networks, Network Hardware, Network Software, Network Topologies (Star, Ring, Bus, Mesh), Network Classifications, Network Protocol, Layered Network Architecture, Overview of OSI Reference Model, Overview of TCP/IP protocol suite, Data
- UNIT II** Communication Fundamentals and Techniques, Analog and Digital Signal, Data-rate limits, Multiplexing Techniques - FDM, TDM, Transmission Media, Networks Switching Techniques and Access Mechanisms, Circuit Switching, Packet Switching-Connectionless Datagram Switching, Connection-Oriented, Virtual Circuit Switching.
- UNIT III** Dial-up Modems, Digital Subscriber Line, Cable TV for data transfer, CSMA/CD protocols, Ethernet LANs, Connecting LAN and Back-bone Networks - Repeaters, Hubs, Switches, Bridges, Router and Gateways, Routing protocols, IP Addressing.
- UNIT IV** Transport Layer Functions and Protocols, Error and flow control, Connection establishment and release, Three-way handshaking, Overview of Application Layer Protocol, DNS,
- UNIT V** Overview of WWW & HTTP Protocols, Electronic Mail.

Recommended Books and References:

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM ,2007.
2. A. S. Tanenbaum: Computer Networks, Fourth edition, PHI , 2002

CORE 7 (CSC 3.32)
COMPUTER NETWORKS LAB

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate and implement stop and wait protocol for noisy channel.
3. Simulate and implement go back n sliding window protocol.
4. Simulate and implement selective repeat sliding window protocol.
5. Simulate and implement distance vector routing algorithm
6. Simulate and implement Dijkstra algorithm for shortest path routing.

SEMESTER – IV

CORE 8 (CSC 4.11)

DESIGN AND ANALYSIS OF ALGORITHMS

Theory Credit: 4

Teaching Hours: 60

- UNIT I Introduction:** Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.
- UNIT II Algorithm Design Techniques:** Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.
- UNIT III Sorting and Searching Techniques:** Elementary sorting techniques–Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis;
- UNIT IV** Decision Trees Red-Black Trees Amortized analysis
- UNIT V Graphs:** Graph Algorithms–Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees.

Recommended Books and References:

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2. Sarabasse & A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999

CORE 8 (CSC 4.12)

DESIGN AND ANALYSIS OF ALGORITHMS LAB

1. i. Implement Insertion Sort (The program should report the number of comparisons)
ii. Implement Merge Sort (The program should report the number of comparisons)
2. Implement Heap Sort (The program should report the number of comparisons)
3. Implement Randomized Quick sort (The program should report the number of comparisons)
4. Implement Radix Sort
5. Create a Red-Black Tree and perform following operations on it:
 - i. Insert a node
 - ii. Delete a node
 - iii. Search for a number & also report the color of the node containing this number.
6. Write a program to determine the LCS of two given sequences
7. Implement Breadth-First Search in a graph
8. Implement Depth-First Search in a graph
9. Write a program to determine the minimum spanning tree of a graph

For the algorithms at S.No 1 to 3 test run the algorithm on 100 different inputs of sizes varying from 30 to 1000. Count the number of comparisons and draw the graph. Compare it with a graph of $n \log n$.

**CORE 9 (CSC 4.21)
SOFTWARE ENGINEERING THEORY**

Theory Credit: 4

Teaching Hours: 60

- UNIT I Introduction:** The Evolving Role of Software, Software Characteristics, Changing Nature of Software, Software Engineering as a Layered Technology, Software Process Framework, Framework and Umbrella Activities, Process Models, Capability Maturity Model Integration (CMMI).
- UNIT II Requirement Analysis:** Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components of SRS.
- UNIT III Quality Management:** Quality Concepts, Software Quality Assurance, Software Reviews, Metrics for Process and Projects.
- UNIT IV Design Engineering:** Design Concepts, Architectural Design Elements, Software Architecture, Data Design at the Architectural Level and Component Level, Mapping of Data Flow into Software Architecture, Modeling Component Level Design.
- UNIT V Testing Strategies & Tactics:** Software Testing Fundamentals, Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System testing, Black-Box Testing, White-Box Testing and their type, Basis Path Testing.

Recommended Books and References:

1. R.S. Pressman, Software Engineering: A Practitioner’s Approach (7th Edition), McGraw- Hill, 2009.
2. P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa Publishing House, 2003.
3. K.K. Aggarwal and Y. Singh, Software Engineering (2nd Edition), New Age International Publishers, 2008.
4. I. Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.
5. D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005.
6. R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India, 2004.

**CORE 9 (CSC 4.22)
SOFTWARE ENGINEERING LAB**

SL NO.	Practical Title
1	<ul style="list-style-type: none"> • Problem Statement, • Process Model
2	Requirement Analysis: <ul style="list-style-type: none"> • Creating a Data Flow • Data Dictionary, Use Cases
3	Project Management: Computing FP Effort <ul style="list-style-type: none"> • Schedule, Risk Table, Timeline chart
4	Design Engineering: <ul style="list-style-type: none"> • Architectural Design

	<ul style="list-style-type: none"> • Data Design, Component Level Design
5	Testing: <ul style="list-style-type: none"> • Basis Path Testing

Sample Projects:

1. **Criminal Record Management:** Implement a criminal record management system for jailers, police officers and CBI officers
2. **DTC Route Information:** Online information about the bus routes and their frequency and fares
3. **Car Pooling:** To maintain a web based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation system
8. Automatic Internal Assessment System
9. Parking Allocation System
10. Wholesale Management System

CORE 10 (CSC 4.31)

DATABASE MANAGEMENT SYSTEMS

Theory Credit: 4

Teaching Hours: 60

UNIT I 1. Introduction: Characteristics of database approach, data models, database system architecture and data independence.

UNIT II Entity Relationship (ER) Modeling: Entity types, relationships, constraints.

UNIT III Relation data model: Relational model concepts, relational constraints, relational algebra, SQL queries

UNIT IV Database design: Mapping ER/EER model to relational database, functional dependencies, Lossless decomposition, Normal forms (upto BCNF).

UNIT V File Structure and Indexing: Operations on files, File of Unordered and ordered records, overview of File organizations, Indexing structures for files (Primary index, secondary index, clustering index), Multilevel indexing using B and B+ trees.

Recommended Books and References:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

CORE 10 (CSC 4.32)
DATABASE MANAGEMENT SYSTEMS LAB PRACTICAL

1. Create and use the following database schema to answer the given queries.

EMPLOYEE Schema

Field	Type	NULL	KEY	DEFAULT
Eno	Char(3)	NO	PRI	NIL
Ename	Varchar(50)	NO		NIL
Job_type	Varchar(50)	NO		NIL
Manager	Char(3)	Yes	FK	NIL
Hire_date	Date	NO		NIL
Dno	Integer	YES	FK	NIL
Commission	Decimal(10,2)	YES		NIL
Salary	Decimal(7,2)	NO		NIL

DEPARTMENT Schema

Field	Type	NULL	KEY	DEFAULT
Dno	Integer	No	PRI	NULL
Dname	Varchar(50)	Yes		NULL

Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.
4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than \$2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of \$1500 and \$2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is _A'.
14. Query to display Name of all employees either have two _R's or have two _A's in their name and are either in Dept No = 30 or their Manger's Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.

SEMESTER – V

CORE 11 (CSC 5.11) INTERNET TECHNOLOGIES

Theory Credit: 4

Teaching Hours: 60

- UNIT I** **Java:** Use of Objects, Array and Array List class
- UNIT II** **JavaScript:** Data types, operators, functions, control structures, events and event handling.
- UNIT III** **JDBC:** JDBC Fundamentals, Establishing Connectivity and working with connection interface, Working with statements, Creating and Executing SQL Statements, Working with Result Set Objects.
- UNIT IV** **JSP:** Introduction to Java Server Pages, HTTP and Servlet Basics, The Problem with Servlets, The Anatomy of a JSP Page, JSP Processing, JSP Application Design with MVC, Setting Up the JSP Environment, Implicit JSP Objects, Conditional Processing, Displaying Values, Using an expression to Set an Attribute, Declaring Variables and Methods, Error Handling and Debugging, Sharing Data Between JSP Pages, Requests, and Users, Database Access.
- UNIT V** **Java Beans:** Java Beans Fundamentals, JAR files, Introspection, Developing a simple Bean, Connecting to DB

Recommended Books and References:

1. Ivan Bayross, Web Enabled Commercial Application Development Using Html, Dhtml, javascript, Perl Cgi , BPB Publications, 2009.
2. Cay Horstmann, BIG Java, Wiley Publication, 3rd Edition., 2009
3. Herbert Schildt, Java 7, The Complete Reference, , 8th Edition, 2009.
4. Jim Keogh, The Complete Reference J2EE, TMH, , 2002.
5. O'Reilly, Java Server Pages, Hans Bergsten, Third Edition, 2003.

CORE (CSC 5.12) INTERNET TECHNOLOGIES LAB

Create event driven program for following:

1. Print a table of numbers from 5 to 15 and their squares and cubes using alert.
2. Print the largest of three numbers.
3. Find the factorial of a number n.
4. Enter a list of positive numbers terminated by Zero. Find the sum and average of these numbers.
5. A person deposits Rs 1000 in a fixed account yielding 5% interest. Compute the amount in the account at the end of each year for n years.
6. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list

CORE 12 (CSC 5.21)
THEORY OF COMPUTATION

Theory Credit: 4

Teaching Hours: 60

Tutorial Credit: 1

UNIT I **Languages:** Alphabets, string, language, Basic Operations on language, Concatenation, KleeneStar

UNIT II **Finite Automata and Regular Languages:** Regular Expressions, Transition Graphs, Deterministic and non-deterministic finite automata, NFA to DFA Conversion, Regular languages and their relationship with finite automata, Pumping lemma and closure properties of regular languages.

UNIT III **Context Free Languages:** Context free grammars, parse trees, ambiguities in grammars and languages, Pushdown automata (Deterministic and Non-deterministic), Pumping Lemma, Properties of context free languages, normal forms.

UNIT IV **Turing Machines and Models of Computation**
1. RAM, Turing Machine as a model of computation, Universal Turing Machine, Language
2. acceptability, decidability, halting problem,

UNIT V Recursively enumerable and recursive languages, insolvability problems.

Recommended Books and References:

1. Daniel I.A. Cohen, Introduction to computer theory, John Wiley, 1996
2. Lewis & Papadimitriou, Elements of the theory of computation, PHI 1997.
3. Hopcroft, Aho, Ullman, Introduction to Automata theory, Language & Computation –3rd Edition, Pearson Education. 2006
4. P. Linz, An Introduction to Formal Language and Automata 4th edition Publication Jones Bartlett, 2006

SEMESTER – VI

CORE 13 (CSC 6.11) ARTIFICIAL INTELLIGENCE

Theory Credit: 4

Teaching Hours: 60

- UNIT I Introduction:** Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.
- UNIT II Problem Solving and Searching Techniques:** Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.
- UNIT III Knowledge Representation:** Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG)
- UNIT IV Dealing with Uncertainty and Inconsistencies:** Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.
- UNIT V Understanding Natural Languages:** Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.

Recommended Books and References:

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.
2. Russell & Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2nd edition, 2005.
3. Rich & Knight, Artificial Intelligence – Tata McGraw 2nd Hill,
4. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing edition, 1991. House, 3rd edition, 2001.
5. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3rd edition, 2000.

CORE 12 (CSC 6.12) ARTIFICIAL INTELLIGENCE LAB

1. Write a prolog program to calculate the sum of two numbers.
2. Write a prolog program to find the maximum of two numbers.
3. Write a prolog program to calculate the factorial of a given number.
4. Write a prolog program to calculate the nth Fibonacci number.
5. Write a prolog program, insert_nth(item, n, into_list, result) that asserts that result is the list into_list with item inserted as the nth element into every list at all levels.
6. Write a Prolog program to remove the Nth item from a list.
7. Write a Prolog program, remove_nth(Before, After) that asserts the After list is the Before list with the removal of every nth item from every list at all levels.
8. Write a Prolog program to implement append for two lists.

9. Write a Prolog program to implement palindrome(List).
10. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
11. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
12. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement reverse(List,ReversedList) that reverses lists.
15. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.
16. Write a Prolog program to implement GCD of two numbers.
17. Write a prolog program that implements Semantic Networks/Frame Structures.

**CORE 14 (CSC 6.21)
COMPUTER GRAPHICS**

Theory Credit: 4

Teaching Hours: 60

UNIT I Introduction: Basic elements of Computer graphics, Applications of Computer Graphics.

UNIT II Graphics Hardware: Architecture of Raster and Random scan display devices, input/output devices.

UNIT III Fundamental Techniques in Graphics: Raster scan line, circle and ellipse drawing, thick primitives, Polygon filling, line and polygon clipping algorithms, 2D and 3D Geometric Transformations, 2D and 3D Viewing Transformations (Projections- Parallel and Perspective), Vanishing points.

UNIT IV Geometric Modeling: Representing curves & Surfaces.

UNIT V Hidden surface elimination. Illumination and shading models. Basic color models and Computer Animation

Recommended Books and References:

1. J.D. Foley, A. Van Dam, S. K. Van Dam, J. D. Foley, Computer Graphics Principles & Practice 2nd edition Publication Addison Wesley 1990.
2. D. Hearn, Baker: Computer Graphics, Prentice Hall of India 2008.
3. D.F. Rogers Procedural Elements for Computer Graphics, McGraw Hill 1997.
4. D.F. Rogers, Adams Mathematical Elements for Computer Graphics, McGraw Hill 2nd edition 1989.

CORE 14 (CSC 6.22)**COMPUTER GRAPHICS LAB**

1. Write a program to implement Bresenham's line drawing algorithm.
2. Write a program to implement mid-point circle drawing algorithm.
3. Write a program to clip a line using Cohen and Sutherland line clipping algorithm.
4. Write a program to clip a polygon using Sutherland Hodgeman algorithm.
5. Write a program to apply various 2D transformations on a 2D object (use homogenous coordinates).
6. Write a program to apply various 3D transformations on a 3D object and then apply parallel and perspective projection on it.
7. Write a program to draw Hermite/Bezier curve.

DISCIPLINE SPECIFIC ELECTIVES

DISCIPLINE SPECIFIC ELECTIVES 1 (CSD 5.11) PHP PROGRAMMING

Theory Credit: 4

- UNIT I Introduction to PHP:** PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database, Editors Basic Syntax, PHP variables PHP with other technologies, scope of PHP etc.) and constants Types of data in PHP, Expressions, scopes of a variable (local, global) PHP Operators: Arithmetic, Assignment, Relational, Logical operators, Bitwise, ternary and MOD operator. PHP operator Precedence and associativity
- UNIT II Handling HTML form with PHP:** GET and Capturing Form Data Redirecting a form after Dealing with multi value fields POST form methods submission
- UNIT III PHP conditional events and Loops:** PHP IF Else conditional Switch case, while, For and Do While Loop statements (Nested IF and Else) Goto, Break, Continue and exit
- UNIT IV PHP Functions:** Function, Need of Function, declaration and calling of a Function PHP Function with arguments, Default Arguments in Function function Scope of Function Global and argument with call by value, call by reference Local
- UNIT V String Manipulation and Regular Expression:** Creating and accessing String, Searching & Replacing String Formatting, joining and splitting String, String Use and advantage of regular expression over inbuilt Related Library functions Use of preg_match(), preg_replace(), preg_split() functions in function regular expression

Recommended Books and References:

1. Steven Holzner, "PHP: The Complete Reference Paperback", McGraw Hill Education (India), 2007.
2. Timothy Boronczyk, Martin E. Psinas, "PHP and MYSQL (Create-Modify-Reuse)", Wiley India Private Limited, 2008.
3. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5", 3rd Edition Paperback, O'reilly, 2014.

DISCIPLINE SPECIFIC ELECTIVES 1 (CSD 5.12) SOFTWARE LAB BASED ON PHP

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.
5. Write a PHP function that checks if a string is all lower case.

6. Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.
Sample string: 'The quick " " brown fox' Expected Output: Thequick""brownfox

DISCIPLINE SPECIFIC ELECTIVES 2 (CSD 5.21)

PROGRAMMING IN VISUAL BASIC

UNIT I Overview Environment: *Overview:* Visual Basic Application Types, Visual Basic Application Components- (Projects, Forms, Controls, Code modules, Class modules, User controls, Property pages)

VB Environment: Menu Bar, Toolbar, Toolbox, Form, Project explorer, Property window, Immediate window, Form layout window. Creating a project, Forms, Naming a project, Saving a project.

Controls: Label control, TextBox control, Command Button, Frames, Option Buttons, Check Boxes, Picture control, Image Control, Shape control, Line control, Timer control, HscrollBar control, VscrollBar control, FileListBox control, DirListBox, DriveListBox control.

List and Menus: List Box control, Combo Box Control, Menu editor.

UNIT II Programming: *Variables:* Data types, Declaring variables, Scope and lifetime of a variable, Examples of variables, Variant datatypes. *Arrays:* Different Array types, Control Array.

Conditional Logic and Looping: If... Then, Select... Case, Do...While, While...Wend, Loop...While, Do...Until, Loop...Until, For... Next, Nested constructs, Exit For/Exit Do, Exit Sub/Exit Function

UNIT III Built in Functions: *Built-in Functions:* String Functions, Date Functions, MsgBox, InputBox, Common Dialog Control.

UNIT IV Data control and Programming: JET database Engine, ADODC, DAO Data Control
Data Entry with ADO: Creating a new ADO project, Adding data, Editing data, Deleting data.

UNIT V Creating Reports: Data Report and crystal Report.

Recommended Books and References:

1. Paul D Sheriff Sheriff, Paul Sheriff Teaches Visual Basic 6, Prentice Hall of India
2. Evangelos Petroutsos, Mastering Visual Basic 6, BPB Publication,
3. Michel C Amundsen and Curties L Smith Smith, Teach Yourself Database Programming with Visual Basic 5 in 21 Days, TechMedia
4. Peter Norton, Peter Norton's Guide to Visual Basic 6, Techmedia

DISCIPLINE SPECIFIC ELECTIVES 2 (CSD 5.22)
VISUAL BASIC (PRACTICAL)

1. Design a form and place a TextBox in it. Call it (assign its name property) txtInput. Place a Command Button and call it cmdExtract. Assign the caption property of the Command button as "Extract". Write a program to extract each digit or letter of a number, word or sentence that is entered in txtInput and display them in a second Text Box called txtOutput, one at a time on the click of a button.
2. Design a form with suitable controls to input a single digit number and write appropriate event handlers to check if the number is automorphic or not. A number is called automorphic if the last digit of the square of the number is same as the number itself. (e.g. 6).
3. Design a form with suitable controls and write appropriate event handlers to take in a string and determine whether the given string is palindrome or not.
4. Design a form with suitable controls and write appropriate event handlers to generate the calendar of a given month. The user must enter the month and the year. Assume that 1st January 1900 was a Monday. Do not use the standard Visual Basic functions to generate the calendar.
5. Write a program to calculate and display the factorial of a given number, using a recursive function.
6. The following information is to be maintained regarding the users of electricity: Name, code and units consumed. Write a program that will take the name and units consumed and hence generate a bill. For the first 20 units cost is 30p/unit, for the next 20 units, 40p/unit, for the rest, 50p/unit. Make provisions for reading, editing and deleting data. Make provisions to keep the rates alterable. Use ADO data control.

DISCIPLINE SPECIFIC ELECTIVES 3 (CSD 6.11)
CLOUD COMPUTING

- UNIT I** Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing
- UNIT II** Cloud Computing Architecture, Comparison with traditional computing architecture (client/server), Services provided at various levels,
- UNIT 3** Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud
- UNIT IV** Service Management in Cloud Computing, Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.
Examples of cloud systems like Google App Engine, Microsoft Azure, AmazonEC2
- UNIT V** Cloud Security, Infrastructure Security, Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing

Recommended Books and References:

1. Cloud Computing – U S Pandey and K Choudhary, S Chand
2. Cloud Computing: Principles and Paradigms - Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley

**DISCIPLINE SPECIFIC ELECTIVES 3 (CSD 6.12)
SOFTWARE LAB BASED ON CLOUD COMPUTING**

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms.
3. Exploring Google cloud **for the following**
 - a) **Storage**
 - b) **Sharing of data**
 - c) Manage your calendar, to-do lists,
 - d) A document editing tool
4. Exploring Microsoft cloud
5. Exploring Amazon cloud

**DISCIPLINE SPECIFIC ELECTIVES 1 (CSD 6.21)
SOFTWARE DEVELOPMENT PROJECT WORK**

Credit: 6

A student has to undertake a software development project work under the guidance of a teacher during the 6th semester. After completion of the project, the student has to submit a project report which will be evaluated by an External Examiner.

SKILL ENHANCEMENT COURSE

SKILL ENHANCEMENT COURSE 1 (CSS 3.11) HTML PROGRAMMING

Theory Credit: 2

- UNIT I Introduction** • The Head, the Body Colors, Attributes Lists, ordered and unordered
- UNIT II Links** Introduction Relative Links, Absolute Links Link Attributes Using the ID Attribute to Link Within a Document
- UNIT III Images** (2L) o Putting an Image on a Page Using Images as Links Putting an Image in the Background
- UNIT IV Tables** (4L) o Creating a Table Table Headers Captions Spanning Multiple Columns Styling Table
- UNIT VI Forms** Basic Input and Attributes Other Kinds of Inputs Styling forms with CSS Where To Go From Here

Recommended Books and References:

1. Introduction to **HTML** and CSS -- O'Reilly, 2010
2. Jon Duckett, HTML and CSS, John Wiely, 2012

SOFTWARE LAB BASED ON HTML

Q.1 Create an HTML document with the following formatting options:

- I. Bold
- II. Italics
- III. Underline
- IV. Headings (Using H1 to H6 heading styles)
- V. Font (Type, Size and Color)
- VI. Background (Colored background/Image in background)
- VII. Paragraph
- VIII. Line Break
- IX. Horizontal Rule
- X. Pre tag

Q.2 Create an HTML document which consists of:

- I. Ordered List
- II. Unordered List
- III. Nested List
- IV. Image

Q.3 Create a form using HTML which has the following types of controls:

- I. Text Box
- II. Option/radio buttons
- III. Check boxes
3. IV. Reset and Submit buttons

Q4 Create an HTML document which implements Internal linking as well as External linking.

SKILL ENHANCEMENT COURSE 2 (CSS 4.11) PROGRAMMING IN MATLAB

- UNIT I** Introduction to Programming: Components of a computer, working with numbers, Machine code, Software hierarchy
- UNIT II** Programming Environment: MATLAB Windows, A First Program, Expressions, Constants, Variables and assignment statement, Arrays.
- UNIT III** Graph Plots: Basic plotting, Built in functions, Generating waveforms, Sound replay, load and save. Procedures and Functions: Arguments and return values, M-files Formatted console input-output, String handling
- UNIT IV** Control Statements: Conditional statements: If, Else, Else-if, Repetition statements: While, for loop
- UNIT VI** Manipulating Text: Writing to a text file, Reading from a text file, Randomising and sorting a list, searching a list. GUI Interface: Attaching buttons to actions, Getting Input, Setting Output

Recommended Books and References:

1. MATLAB: An Introduction with Applications, by Amos Gilat, 2nd edition, Wiley, 2004,
2. C.B. Moler, Numerical Computing with MATLAB, SIAM, 2004.

SOFTWARE LAB BASED ON MATLAB

1. Celsius temperatures can be converted to Fahrenheit by multiplying by 9, dividing by 5, and adding 32. Assign a variable called C the value 37 and implement this formula to assign a variable F the Fahrenheit equivalent of 37 Celsius.
2. supermarket conveyor belt holds an array of groceries. The price of each product (in pounds) is [0.6, 1.2 ,0.5, 1.3]; while the numbers of each product are [3, 2 ,1 ,5]. Use MATLAB to calculate the total bill.
3. The sortrows(x) function will sort a vector or matrix X into increasing row order. Use this function to sort a list of names into alphabetical order.
4. The $-identity$ matrix is a square matrix that has ones on the diagonal and zeros elsewhere. You can generate one with the eye() function in MATLAB. Use MATLAB to find a matrix B, such that when multiplied by matrix $A = \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}$ the identity matrix $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is generated. That is $A*B=I$.
5. Create an array of N numbers. Now find a single MATLAB statement that picks out from that array the 1,4,9,16,...,Nth entries, i.e. those numbers which have indices that are square numbers.
6. Draw a graph that joins the points (0,1), (4,3), (2,0) and (5,-2).