Revised CBCS SYLLABUS

FOR

THREE YEARS UNDER-GRADUATE

COURSE IN

COMPUTER SCIENCE (HONOURS)

(*w.e.f. 2022-23*)

BANKURA UNIVERSITY

BANKURA, WEST BENGAL, 722155

After successful completion of 6 Semesters with Computer Science as Core subject a student should be able to:-

Sl No	Program Outcome	Description
PO 1	Sound domain knowledge in Computer Science	To acquire sound theoretical knowledge of different algorithms and emerging areas in Computer Science.
PO 2	Laboratory skill in Computer Science	To be able to write programs in different programming languages and run them in computer.
PO 3	Academic and scientific endeavour	To help the students in developing academic and scientific endeavour by fostering and nurturing the young talent for proper scientific pursuit.
PO 4	Familiarity with recent developments in a particular field	To be able to upgrade themselves to apply modern theories and approaches to explain all spatial phenomena.
PO 5	Spirit of team work	To support students to co-ordinate with one another as team environment.
Sl No	**Program Specific Outcomes**	**Description**
PSO 1	Critical appreciation of the Subject.	To acquire sound theoretical knowledge on the fundamentals of Computer Science and to apply them in practical and professional situations.
PSO 2	Scientific Attitude	To develop the right scientific temperament compatible with creative inclination.
PSO 3	Technical Skill Development	To create updated knowledge on research methodology and to develop skills in application oriented Computer Science.
PSO 4	Communication Skill	To help students develop effective communication skills with the aid of classroom teaching with interactive discussions, student seminar, written assignments and debates which will enhance employability.
PSO 5	Personality Development	To help students develop personality in their professional and personal lives and thus make them responsible and sincere citizens of the society.
PSO 6	Spirit of Team Work	To encourage students to co-ordinate with one another as team member rather than to try to excel individually.
PSO 7	Basic Human Values	To help the learners to understand human behavioural nature by the study of various texts and mutual interaction among the students inside and outside the class room.
PSO 8	Fundamental Nature of Computer Science	To help students comprehend that Computer Science underpins understanding and progress in almost every sphere of science, technology and industry.

STRUCTURE IN COMPUTER SCIENCE (HONOURS)

SEMESTER – I

Course Code	Course Title	Credit	Marks			No. of Hours		
			I.A.	ESE	Total	Lec.	Tu.	Pr.
SH/CSC/ 101/C-1	Programming Fundamentals Using C	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 102/C-2	Computer System Organization and Architecture	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 103/GE-1	**Introduction to Programming**	6	10	40 (T:25 L:15)	50	4	-	4
ACSHP/10 4/ AECC-1	Environmental Studies	4	10	40	50	4		
Total in Semester - I		**22**	**40**	**160**	**200**			

SEMESTER –II

Course Code	Course Title	Credit	Marks			No. of Hours		
			I.A.	ESE	Total	Lec.	Tu.	Pr.
SH /CSC/ 201/C-3	Object Oriented Programming	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 202/C-4	Discrete Mathematics	6	10	40	50	5	1	-
SH /CSC/ 203/GE-2	Introduction to Database Systems	6	10	40 (T:25 L:15)	50	4	-	4
ACSHP/20 4/ AECC-2	English/Hind/MIL	2	10	40	50	2		
Total in Semester - II		**20**	**40**	**160**	**200**			

SEMESTER –III

Course Code	Course Title	Credit	Marks			No. of Hours		
			I.A.	ESE	Total	Lec.	Tu.	Pr.
SH /CSC/ 301/C-5	Data Structures	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 302/ C-6	Operating Systems	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/303 /C-7	Communication and Computer Networks	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 304/GE-3	**Computer Network and Internet Technologies**	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 305/SEC -1	**Any one of the following** ● **Programming in Python** ● **Unix/ Linux Programming**	2	10	40 (T:25 L:15)	50	1	-	2
Total in Semester - III		**26**	**50**	**200**	**250**			

SEMESTER –IV

Course Code	Course Title	Credit	Marks			No. of Hours		
			I.A.	ESE	Total	Lec.	Tu.	Pr.
SH /CSC/401 /C-8	Algorithm Analysis and Design	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/402 /C-9	Software Engineering Concepts	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/403 /C-10	Database Management Systems	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/404 /GE-4	**Programming in Python**	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 405/SEC-2	**Any one of the following** ● **HTML Programming** ● **PHP Programming**	2	10	40 (T:25 L:15)	50	1	-	2
Total in Semester - IV		**26**	**50**	**200**	**250**			

SEMESTER –V

Course Code	Course Title	Credit	Marks			No. of Hours		
			I.A.	ESE	Total	Lec	Tu	Pr
SH /CSC/ 501/C-11	Web Technologies	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 502/C-12	Computing Theory	6	10	40	50	5	1	-
SH /CSC/ 503/DSE-1	**Any one of the following** ● **Numerical Methods** ● **Operations Research**	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 504/DSE-2	**Any one of the following** ● **Microprocessor** ● **Digital Image Processing**	6	10	40 (T:25 L:15)	50	4	-	4
Total in Semester – V		**24**	**40**	**160**	**200**			

SEMESTER – VI

Course Code	Course Title	Credit	Marks			No. of Hours		
			I.A.	ESE	Total	Lec.	Tu.	Pr.
SH /CSC/ 601/C-13	Artificial Intelligence	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 602/C-14	Computer Graphics	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 603/DSE-3	**Any one of the following** ● **Cryptographic Applications** ● **Data Analytics**	6	10	40 (T:25 L:15)	50	4	-	4
SH /CSC/ 604/DSE-4	**Project Work**	6	10	40 (T:25 L:15)	50	4	-	4
Total in Semester – VI		**24**	**40**	**160**	**200**			

SH= Science Honours CSC = Computer Science, ACSHP= Arts Commerce Science Honours Programme, C= Core Course, AECC= Ability Enhancement Compulsory Course, SEC= Skill Enhancement Course, GE= Generic Elective, DSE= Discipline Specific Elective IA= Internal Assessment, ESE= End-Semester Examination, Lec. =Lecture, Tu.= Tutorial, and Prc.=Practical

CORE COURSES (HONOURS IN COMPUTER SCIENCE)

COMPUTER SCIENCE (C-I): Programming Fundamentals using C

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Learn about basic operations of a computer.
- Develop problem solving skills coupled with top down design principles.
- Become skilled at developing simple algorithms and flow charts.
- Convert the algorithms into simple C programs.
- Develop simple C programs for solving real life problems.

Theory: 60 Lectures

1. Introduction to C (3 Lectures)

History of 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 (get c, get char, put c, put char etc), Formatted and Console I/O (printf(), scanf()), Using Basic Header Files (stdio.h , conio.h etc).

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 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 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.

7. Memory Allocation in C **(3 Lectures)**

Differentiating between static and dynamic memory allocation, use of malloc , calloc and free functions, storage of variables in static and dynamic memory allocation.

8. File I/O, Preprocessor Directives **(5 Lectures)**

Opening and closing a file, Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives, Macros.

Reference Books

1. C Programming, Karnighan,&Ritchie, PHI
2. Herbtz Schildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.2003
3. E Balaguruswamy, " Programming with C", Tata McGraw-Hill Education, 2008.
4. Programming through C, Richard Johnsonbaugh and Martin Kalin, Pearson Education
5. Programming in C, B.S. Gottfried, Sahaum Series.
6. Y Kanetkar, “Let us C”, BPB

COMPUTER SCIENCE LAB (C-I): Programming Fundamentals using C
Lab 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() / calloc() functions 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. Write a menu-driven program to perform following Matrix operations (Use 2-D array implementation) a) Sum b) Difference c) Product d) Transpose
22. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
23. Write a program to retrieve the student information from file created in previous question and print it in following format:

    Roll No. Name Marks
24. Copy the contents of one text file to another file, after removing all whitespaces.
25. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
26. 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.

COMPUTER SCIENCE (C-II): Computer System Organization and Architecture

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Learn about function and design of various components of a computer.
- Become skilled at developing different types of combinational and sequential circuits.
- Learn about the working principle of interfaces between software and hardware.
- Learn about the working principle of central processing unit of a Computer.
- Learn about the internal structure of different types of memory used in a computer.
- To make students know the different ways of communicating with I/O devices and standard I/O interfaces.

Theory: 60 Lectures

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, and 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.

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.

Recommended 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.

COMPUTER SCIENCE LAB (C-II):

(Besides the traditional Hardware Platform use of open source Simulators are also encouraged)

Computer System Organization and Architecture Lab Practical: 60 Lectures

1. Implement X-OR Gate using NAND Gates.
2. Implement X-OR Gate using NOR Gates.
3. Implement Half-Adder using Basic Gates.
4. Implement Half-Adder using NAND Gates.
5. Implement Full-Adder using NAND Gates.
6. Implement Full-Subtractor using NAND Gates.
7. Implement the Function: using Basic Gates.
8. Implement the Function: using Basic Gates.
9. Implement the Function: F = ABC + DEF using IC 7411.
10. Implement a 4-bit Binary Adder using IC 7483.
11. Implement 2 × 1 Multiplexer using Basic Gates.
12. Implement 4 × 1 Multiplexer using NAND Gates.
13. Implement 8 × 1 MUX using IC 74153.
14. Implement a 3-bit Even Parity Checker using Basic Gates.
15. Implement a 2-bit Comparator using Basic Gates.
16. Implement S-R Flip-Flop using NAND Gates.
17. Implement J-K Flip-Flop using NAND Gates.

Computer Science(C-III) Object Oriented Programming

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Learn about Object Oriented Programming.
- Use, write, compile, debug and learn in C++ and Java Programming.
- The practical part of this course will enable the students to develop apps based on Java.
- Develop Web Site and Web Applications.

Theory: 60 Lectures

1. Object-Oriented Programming Overview (4 Lectures)

Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Function/Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

Object-Oriented Programming through C++

2. Using Classes in C++ (10 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.

3. Overview of Function Overloading and Operator Overloading (6 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)

4. Inheritance, Polymorphism and Exception Handling (8 Lectures)

Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions)

Object-Oriented Programming through Java

5. Introduction to Java (4 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),

6. Arrays, Strings and I/O (8 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.

7. Inheritance, Interfaces, Packages (12 lectures)

Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes.

8. Exception Handling, Threading - (15 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.

9. Applets and Event Handling (8 Lectures)

Java Applets: Introduction to Applets, Writing Java Applets, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes.

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

COMPUTER SCIENCE LAB (C-III): Object-Oriented Programming Lab

Practical: 60 Lectures

All Programs are to be run in either C++ or Java or Both wherever applicable

1. Create Matrix class using templates. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
   a) Sum b) Difference c) Product d) Transpose
2. 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).
3. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.
4. 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.
5. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
6. Write a program that show working of different functions of String and String Buffer class like setCharAt(), setLength(), append(), insert(), concat()and equals().
7. 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
8. 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.
9. 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)
10. 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
11. Write a program to show the use of static functions and to pass variable length arguments in a function.
12. 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.
13. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacii series is given ina different file belonging to the same package.

14. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
15. Write a program ―DivideByZero 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.
16. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.
17. 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).
18. Write a program to demonstrate priorities among multiple threads.
19. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).
20. Write a program to create URL object, create a URL Connection using the openConnection () method and then use it examine the different components of the URL and content.
21. 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.
22. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
23. Write a program to demonstrate different keyboard handling events.
24. Write a program to generate a window without an applet window using main() function.
25. Write a program to demonstrate the use of push buttons.

COMPUTER SCIENCE (C-IV): Discrete Mathematics

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Study the objectives and problems related to all branches of computer science.
- Apply mathematical thinking, mathematical proofs, and algorithmic thinking in problem solving.
- Understand the basics of combinatories, and be able to apply the methods in problem solving.
- Understand some basic properties of graphs and related discrete structures, and be able to relate these to practical examples.

Theory: 60 Lectures

1. Introduction: **(15 Lectures)**

Sets - finite and Infinite sets, uncountably Infinite Sets; functions, relations, Properties of Binary Relations, Closure, Partial Ordering Relations; counting - Pigeonhole Principle, Permutation and Combination; Mathematical Induction, Principle of Inclusion and Exclusion.

2. Growth of Functions: **(8 Lectures)**

Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals

3. Recurrences: **(10 Lectures)**

Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem

4. Graph Theory **(15 Lectures)**

Basic Terminology, Models and Types, multigraphs and weighted graphs, Graph Representation, Graph Isomorphism, Connectivity, Euler and Hamiltonian Paths and Circuits, Planar Graphs, Graph Coloring, Trees, Basic Terminology and properties of Trees, Introduction to Spanning Trees

5. Prepositional Logic **(12 Lectures)**

Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory

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, Sixth 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

COMPUTER SCIENCE (C-IV): Discrete Mathematics Tutorial**: 15 lectures**

COMPUTER SCIENCE (C-V): Data Structures

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Construct and analysis various data structures and abstract data types including lists, stacks, queues, trees, and graphs.
- Perform basic operations of insert, delete, search etc on data structures like tree, Linked List, stacks queues etc.
- Choose a data structure to suitably model any data used in any applications.
- Write programs using different data structures like hash tables, linked lists, trees, graphs etc.

Theory: 60 Lectures

1. **Arrays** **(5 Lectures)**
   Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation)

2. **Stacks** **(5 Lectures)**
   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

3. **Linked Lists** **(10 Lectures)**
   Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists

4. **Queues** **(5 Lectures)**
   Array and Linked representation of Queue, De-queue, Priority Queues

5. **Recursion** **(5 lectures)**
   Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

6. **Trees** **(20 Lectures)**
   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).

7. **Searching and Sorting** **(5 Lectures)**
   Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques

8. **Hashing** **(5 Lectures)**
   Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collusion by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing Function

Reference Books:

1. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.
2. SartajSahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
3. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "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.

COMPUTER SCIENCE LAB (C-V): Data Structures Lab

Practical: 60 Lectures

---

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) 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 Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.

COMPUTER SCIENCE (C-VI): Operating Systems

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Acquire knowledge in the objectives of operating systems.
- Know how operating systems are related to computer hardware, what functionalities are provided to users, and what the major components in operating systems are.
- Get familiarization with LINUX system calls for process management and inter-process communication.
- Perform experiments on process scheduling and other operating system tasks through simulation.
- Have a basic knowledge about multithreading.
- Understand concepts of memory management including virtual memory.
- Understand issues related to file system interface and implementation, disk management.

Theory: 60 Lectures

1. Introduction (10 Lectures)

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.

2. Operating System Organization (6 Lectures)

Processor and user modes, kernels, system calls and system-programs.

3. Process Management (20 Lectures)

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.

4. Memory Management (10 Lectures)

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

5. File and I/O Management (10 Lectures)

Directory structure, file operations, file allocation methods, device management.

6. Protection and Security (4 Lectures)

Policy mechanism, Authentication, Internal access Authorization.

Recommended Books:

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 Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

COMPUTER SCIENCE LAB (C-VI): Operating Systems Lab

Practical: 60 Lectures

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 behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behaviour 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

COMPUTER SCIENCE (C-VII): Communications and Computer Networks

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Acquire the computer networking knowledge as well as the existing connectivity technologies.
- Establish a solid knowledge of the layered approach.
- Acquire the knowledge of the basic protocols involved in wired/wireless communication process.
- Get practical approaches to Ethernet/Internet networking: networks are assembled, and experiments are made to understand the layered architecture and how do some important protocols work?

Theory: 60 Lectures

1. Introduction to Computer Networks (8 Lectures)

Network definition; network topologies; network classifications; network protocol; layered network architecture; overview of OSI reference model; overview of TCP/IP protocol suite.

2. Data Communication Fundamentals and Techniques (10 Lectures)

Analog and digital signal; data-rate limits; digital to digital line encoding schemes; pulse code modulation; parallel and serial transmission; digital to analog modulation-; multiplexing techniques- FDM, TDM; transmission media.

3. Networks Switching Techniques and Access mechanisms (10 Lectures)

Circuit switching; packet switching- connectionless datagram switching, connection-oriented virtual circuit switching; dial-up modems; digital subscriber line; cable TV for data transfer.

4. Data Link Layer Functions and Protocol (10 Lectures)

Error detection and error correction techniques; data-link control- framing and flow control; error recovery protocols- stop and wait ARQ, go-back-n ARQ; Point to Point Protocol on Internet.

5. Multiple Access Protocol and Networks (5 Lectures)

CSMA/CD protocols; Ethernet LANS; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways;

6. Networks Layer Functions and Protocols (6 Lectures)

Routing; routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols.

7. Transport Layer Functions and Protocols (6 Lectures)

Transport services- error and flow control, Connection establishment and release- three way handshake;

8. Overview of Application layer protocol (5 Lectures)

Overview of DNS protocol; overview of WWW &HTTP protocol.

Reference Books

1. B. A. Forouzan: Data Communications and Networking, Fourth edition, THM ,2007.
2. William Stallings:Data and Computer Communications, Eight Edition, Pearson.
3. A. S. Tanenbaum: Computer Networks**,** Fourth edition**,** PHI , 2002

COMPUTER SCIENCE LAB (C-VII): Computer Networks

Lab Practical: 60 Lectures

---

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.

COMPUTER SCIENCE (C-VIII): Algorithm Analysis and Design

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Know the structure of an algorithm.
- Design algorithms to solve different types of problems in the branch of computer science and information technology.
- To learn how to analyse algorithms and estimate their worst-case and average case behaviour (in easy cases)
- Design algorithm which refers to a method or a mathematical process for problem-solving and engineering algorithms.

Theory: 60 Lectures

1. Introduction (5 Lectures)

Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.

2. Algorithm Design Techniques (8 Lectures)

Iterative techniques, Divide and Conquer, Dynamic Programming, Greedy Algorithms.

3. Sorting and Searching Techniques (20 Lectures)

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;

4. Lower Bounding Techniques (5 Lectures)

Decision Trees

5. Balanced Trees (7 Lectures)

Red-Black Trees

6. Advanced Analysis Technique (5 Lectures)

Amortized analysis

7. Graphs (5 Lectures)

Graph Algorithms–Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees.

8. String Processing (5 Lectures)

String Matching, KMP Technique

Recommended Books:

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

COMPUTER SCIENCE LAB (C-VIII): Algorithm Analysis and Design

Lab Practical: 60 Lectures

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
10. 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 nlogn.

COMPUTER SCIENCE (C-IX): Software Engineering Concepts

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Get basic knowledge and understanding of the analysis and design of complex systems.
- Develop various theoretical implementations of software with the knowledge of software engineering. This can help to create new software.
- Learn and implement different types of application software. They can build different types of software with the theoretical help of software engineering.
- Work as an effective member or leader of software engineering teams.
- To manage time, processes and resources effectively by prioritizing competing demands to achieve their goals.
- Identify and analyze the common threats in each domain.

Theory: 60 Lectures

1. Introduction (8 Lectures)

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).

2. Requirement Analysis (10 Lectures)

Software Requirement Analysis, Initiating Requirement Engineering Process, Requirement Analysis and Modeling Techniques, Flow Oriented Modeling, Need for SRS, Characteristics and Components for SRS

3. Software Project Management (8 Lectures)

Estimation in Project Planning Process, Project Scheduling

4. Risk Management (8 Lectures)

Software Risks, Risk Identification, Risk Projection and Risk Refinement, RMMM Plan

5. Quality Management (8 Lectures)

Quality Concepts, Software Quality Assurance, Software Review, Metrics for Process and Project

6. Design Engineering (8 Lectures)

Design Concepts, Architectural Design Elements, Software Architecture, Design at the Architectural Level and Component Level, Mapping of Data Flow into Software Architecture, Modeling Component Level Design.

7. Testing Strategies (8 Lectures)

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:

1. R S Pressman, Software Engineering; A Practitioner's Approach (7th Edition), MGH, 2009

2. P Jalote, An Integrated Approach to Software Engineering (2nd Edition), NPH, 2003

3. R. Mall, Fundamentals of Software Engineering (2nd Edition), PHI, 2004

COMPUTER SCIENCE LAB (C-IX): Software Engineering Concepts Lab
Practical: 60 Lectures

S. No	Practical Title
1.	Problem Statement, Process Model
2.	Requirement Analysis: • Creating a Data Flow • Data Dictionary, • Use Cases
3.	Project Management: • Computing FP • Effort, Schedule, Risk Table, Timeline Chart
4.	Design Engineering: • Architectural Design • Data Design, Component Level Design
5.	Testing: Basic 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
6. Examination and Result computation system
7. Automatic Internal Assessment System
8. Parking Allocation System
9. Wholesale Management System

COMPUTER SCIENCE (C-X): Database Management Systems

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Gain knowledge of database systems and database management systems software
- Work with a huge database. Through database management system they can work with any real life database.
- Work in different databases using PL-SQL. They can create, delete, and update the database in this class.
- Demonstrate an understanding of normalization theory and apply such knowledge to the normalization of a database.

Theory: 60 Lectures

1. **Introduction:** **(6 Lectures)**

Characteristics of Database Approach, Data Models, Database System Architecture and Data Independence

2. Entity Relationship (ER) Modeling **(8 Lectures)**

Entity Types, Relations, Constraints

3. Relational Data Model **(20 Lectures)**

Relational Model Concepts, Relational Constraints, Relational Algebra, SQL Queries

4. Database Design **(15 Lectures)**

Mapping ER model to relational database, functional dependencies, Lossless decomposition, and Normal forms (up to BCNF).

5.Transaction Processing **(3 Lectures)**

ACID properties, concurrency control

6.File Structure and Indexing **(8 Lectures)**

Operations on File, Unordered and Ordered Records, Overview of File Organizations, Indexing Structures for Files, B and B^+ Trees

Books Recommended:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6^{th} E, PE, 2010
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3^{rd} E, MGH, 2002
3. A. Silberschatz, H.F. Korth, S. Sudarshan, Database System Concepts 6^{th} E, PE

COMPUTER SCIENCE LAB (C-X): Database Management Lab

Practical: 60 Lectures

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
Location	Varchar(50)	Yes	New Delhi

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%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 * Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with _J‘, ‘A‘ and _M‘.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an _A‘ in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger‘s Name and the Manager‘s employee no; along with the Employees‘ Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (*) signifies $100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees

30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a _T‘.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department.

Practical Evaluation: Experiment-10, Laboratory Note Book-2, Viva voce-3.

COMPUTER SCIENCE (C-XI): Web Technologies

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Design different types of Client side and Server side applications.
- Design Web-enabled applications using JavaScript Programming, Java Server Pages and Java Database Connectivity.
- Work with Java Beans.
- Can learn and implement different applications like stand alone applications, web applications etc.

Theory: 60 Lectures

1. Java **(5 lectures)**

Use of Objects, Array and Array List class

2. JavaScript **(15 lectures)**

Data types, operators, functions, control structures, events and event handling.

3. JDBC **(10 lectures)**

JDBC Fundamentals, Establishing Connectivity and working with connection interface, Working with statements, Creating and Executing SQL Statements, Working with Result Set Objects.

4. JSP **(20 lectures)**

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.

5. Java Beans **(10 lectures)**

Java Beans Fundamentals, JAR files, Introspection, Developing a simple Bean, Connecting to DB

Recommended Books:

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.

COMPUTER SCIENCE LAB (C-XI): Web Technologies Lab

Practical: 60 Lectures

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.

COMPUTER SCIENCE (C-XII): Computing Theory

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Explain the various concepts of Automata theory and formal languages.
- Design various theoretical implementations associated with computation theory.
- They can solve different problems of machine automaton.
- Apply their understanding of key notions through complex problem solving.

Theory: 60 Lectures

1. Languages: (8 Lectures)

Alphabets, String, Language, Basic Operations on Language, Concatenation, Kleene Star

2. Finite Automata and Regular Languages (20 Lectures)

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.

3. Context free languages (17 Lectures)

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.

4.Turing Machines and Models of Computations (15 Lectures)

RAM, Turing Machine as a model of computation, Universal Turing Machine, Language acceptability, decidability, halting problem, Recursively enumerable and recursive languages, un-solvability problems.

Recommended Books:

1. Daniel I.A.Cohen, Introduction to computer theory, John Wiley,1996
2. Lewis & Papadimitriou, Elements of the theory of computation, PHI 1997.
3. Hoperoft, 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

COMPUTER SCIENCE (C-XII): Computing Theory
Tutorial: 15 Lectures

COMPUTER SCIENCE (C-XIII): Artificial Intelligence

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Learn about various AI based problem solving and searching algorithms.
- Learn about different knowledge representation techniques.
- Solve basic AI problems using prolog programming.

Theory: 60 Lectures

1. Introduction (06 Lectures)

Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.

2. Problem Solving and Searching Techniques (20 Lectures)

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.

3. Knowledge Representation (20 Lectures)

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs.

Programming in Logic (PROLOG)

4. Dealing with Uncertainty and Inconsistencies (08 Lectures)

Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

5. Understanding Natural Languages (06 Lectures)

Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.

BOOKS RECOMMENDED:

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 Hill, 2nd edition, 1991.
4. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3rd edition, 2001.
5. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3rd edition, 2000.

COMPUTER SCIENCE LAB (C-XIII): Artificial Intelligence Lab

Practical: 60 Lectures

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 n'th 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 n'th 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.

COMPUTER SCIENCE (C-XIV): Computer Graphics

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Learn about functioning of different output devices like monitor, printers etc.
- Understand different 2D and 3D graphics objects generating algorithms.
- Design various graphics effects using computer in the laboratory.

Theory: 60 Lectures

1. Introduction (5 Lectures)

Basic elements of Computer graphics, Applications of Computer Graphics.

2. Graphics Hardware (8 Lectures)

Architecture of Raster and Random scan display devices, input/output devices.

3. Fundamental Techniques in Graphics (22 Lectures)

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.

4. Geometric Modeling (10 Lectures)

Representing curves & Surfaces.

5. Visible Surface determination (8 Lectures)

Hidden surface elimination.

6. Surface rendering (7 Lectures)

Illumination and shading models. Basic color models and Computer Animation.

Books Recommended:

1. J.D.Foley, A.Van Dan, Feiner, Hughes 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.

COMPUTER SCIENCE LAB (C-XIV): Computer Graphics Lab

Practical: 60 Lectures

---

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.

Practical Evaluation: Experiment-10, Laboratory Note Book-2, Viva voce-3.

Discipline Specific Elective Papers COMPUTER SCIENCE: (Credit: 06 each)
(4 papers to be selected) – DSE 1 – 4

I. Numerical Methods (DSE-I)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Solve various types of Numerical or Mathematical problems.
- Implement various numerical methods with high accuracy through programming languages.
- Represent statistical data through graphs.

Theory: 60 Lectures

1. Floating point representation and computer arithmetic, Significant digits, Errors: Round-off error, Local truncation error, Global truncation error, Order of a method, Convergence and terminal conditions, Efficient computations 7 **Lectures**

2. Bisection method, Secant method, Regula−Falsi method Newton−Raphson method, Newton‘s method for solving nonlinear systems Gauss elimination method (with row pivoting) and Gauss−Jordan method, Gauss Thomas method for tridiagonal systems **10 Lectures**

3. Iterative methods: Jacobi and Gauss-Seidel iterative methods Interpolation: Lagrange‘s form and Newton‘s form **7 Lectures**

4. Finite difference operators, Gregory Newton forward and backward differences Interpolation Piecewise polynomial interpolation: Linear interpolation, Cubic spline interpolation (only method), Numerical differentiation: First derivatives and second order derivatives, Richardson extrapolation **12 Lectures**

5. Numerical integration: Trapezoid rule, Simpson‘s rule (only method), Newton−Cotes open formulas **8 Lectures**

6. Extrapolation methods: Romberg integration, Gaussian quadrature, Ordinary differential equation: Euler‘s method, Modified Euler‘s methods: Heun method and Mid- point method, Runge-Kutta second methods: Heun method without iteration, Mid-point method and Ralston‘s method **12 Lectures**

7. Classical 4^{th} order Runge-Kutta method, Finite difference method for linear ODE **4 Lectures**

REFERNCE BOOKS:

[1] Laurence V. Fausett, Applied Numerical Analysis, Using MATLAB, Pearson, 2/e (2012)
[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)

Numerical Methods Lab

Practical: 60 lectures

1. Find the roots of the equation by bisection method.
2. Find the roots of the equation by secant/Regula−Falsi method.
3. Find the roots of the equation by Newton‘s method.
4. Find the solution of a system of nonlinear equation using Newton‘s method.
5. Find the solution of tridiagonal system using Gauss Thomas method.
6. Find the solution of system of equations using Jacobi/Gauss-Seidel method.
7. Find the cubic spline interpolating function.
8. Evaluate the approximate value of finite integrals using Gaussian/Romberg integration.
9. Solve the boundary value problem using finite difference method.

II. Operations Research (DSE-I)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Impart knowledge in concepts and tools of Operations Research.
- Understand mathematical models used in Operations Research.
- Develop linear programming (LP) models for solving different real life problems.
- Apply the techniques constructively to make effective business decisions.

Theory: 60 Lectures

1. **Introduction to Operational Research (OR):** Origin & Development, Different Phases of OR study, Methodology of OR, Scope and Limitations of OR, Applications of OR. **10 Lectures**
2. **Linear Programming:** Linearly independent / dependent vectors, Basis, Convex sets, Extreme points. Graphical method. Simplex method, Artificial variable techniques- Two Phase Method; M-Charnes Method, Special cases in LPP. **20 Lectures**
3. **Duality:** Definition of the dual problem, Primal-dual relationships, Economic Interpretation of Duality, Dual simplex Method. **20 Lectures**
4. **Sensitivity analysis:** Changes in cost and resource vector. **10 Lectures**

Reference Books

1. G. Hadley: Linear Programming. Narosa, 2002 (reprint).
2. A. Ravindran, D. T. Phillips and James J. Solberg: Operations Research- Principles and Practice, John Wiley & Sons, 2005.
3. Hamdy A. Taha: Operations Research-An Introduction, Prentice Hall, 8th Edition,2008.
4. F.S. Hillier. G.J. Lieberman: Introduction to Operations Research- Concepts andCases, 9th Edition, Tata McGraw Hill. 2010.

Operations Research Lab
Practical: 60 Lectures

[1] To solve Linear Programming Problem using Graphical Method with Unbounded solution Infeasible solution Alternative or multiple solutions.
[2] Solution of LPP with simplex method.
[3] Problem solving using M-Charnes method.
[4] Problem solving using Two Phase method.
[5] Illustration of following special cases in LPP using Simplex method Unrestricted variables
Unbounded solution Infeasible solution Alternative or multiple solution
[6] Problems based on Dual simplex method.
[7] Problems based on sensitivity analysis.

III. Microprocessor (DSE-II)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Study the functional blocks of Microprocessor.
- Demonstrate the internal architecture (Hardware model) as well as Programming model.
- Get the details knowledge of interfacing.
- Understand Assembly Language Programming.
- Perform different experimental works using Assembly Language.

Theory: 60 Lectures

All concepts should be studied in the context of the Intel 8085 Microprocessor.

Microprocessor architecture: Internal architecture, system bus architecture, memory and I/O interfaces. **20L**

Microprocessor programming: Register Organization, instruction formats, assembly language programming. **20L**

Interfacing: Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, display, timer, interrupt controller, DMA controller, video controllers, communication interfaces. **20L**

Recommended Books:

1. Barry B. Brey : The Intel Microprocessors : Architecture, Programming and Interfacing. Pearson Education, Sixth Edition,2009.
2. Ramesh Gaonkar: Microprocessor Architecture, Programming and Applications with the 8085.

Microprocessor Lab

(Besides the traditional Hardware Platform use of open source Simulators are also encouraged)

Practical: 60 Lectures

All programs should be developed for the Intel 8085 Microprocessor.

ASSEMBLY LANGUAGE PROGRAMMING

1. Write a program for 16-bit binary division and multiplication
2. Write a program for 16-bit BCD addition and subtraction
3. Write a program for linear search and binary search.
4. Write a program to add and subtract two arrays
5. Write a program for binary to ascii conversion
6. Write a program for ascii to binary conversion
7. Write a program for HCF and LCM of two given integers.
8. Write a program for Bubble sort of an array of numbers.

IV. Digital Image Processing (DSE-II)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Understand the need for image transforms different types of image transforms and their properties.
- Develop any image processing application.
- Learn different techniques employed for the enhancement of images.

Theory: 60 Lectures

1. Introduction (6 Lectures)

Light, Brightness adaption and discrimination, Pixels, coordinate conventions, Imaging Geometry, Perspective Projection, Spatial Domain Filtering, sampling and quantization.

2. Spatial Domain Filtering (7 Lectures)

Intensity transformations, contrast stretching, histogram equalization, Correlation and convolution, Smoothing filters, sharpening filters, gradient and Laplacian.

3. Filtering in the Frequency domain (8 Lectures)

Hotelling Transform, Fourier Transforms and properties, FFT (Decimation in Frequency and Decimation in Time Techniques), Convolution, Correlation, 2-D sampling, Discrete Cosine Transform, Frequency domain filtering.

4. Image Restoration (8 Lectures)

Basic Framework, Interactive Restoration, Image deformation and geometric transformations, image morphing, Restoration techniques, Noise characterization, Noise restoration filters, Adaptive filters, Linear, Position invariant degradations, Estimation of Degradation functions, Restoration from projections.

5. Image Compression (10 Lectures)

Encoder-Decoder model, Types of redundancies, Lossy and Lossless compression, Entropy of an information source, Shannon's 1st Theorem, Huffman Coding, Arithmetic Coding, Golomb Coding, LZW coding, Transform Coding, Sub-image size selection, blocking artifacts, DCT implementation using FFT, Run length coding, FAX compression (CCITT Group-3 and Group- 4), Symbol-based coding, JBIG-2, Bit-plane encoding, Bit-allocation, Zonal Coding, Threshold Coding, JPEG, Lossless predictive coding, Lossy predictive coding, Motion Compensation

6. Wavelet based Image Compression (5 Lectures)

Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform (DWT), Continuous Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding, Digital Image Watermarking.

7. Morphological Image Processing (7 Lectures)

Basics, SE, Erosion, Dilation, Opening, Closing, Hit-or-Miss Transform, Boundary Detection, Hole filling, Connected components, convex hull, thinning, thickening, skeletons, pruning, Geodesic Dilation, Erosion, Reconstruction by dilation and erosion.

8. Image Segmentation (9 Lectures)

Boundary detection based techniques, Point, line detection, Edge detection, Edge linking, local processing, regional processing, Hough transform, Thresholding, Iterative thresholding, Otsu's method, Moving averages, Multivariable thresholding, Region-based segmentation, Watershed algorithm, Use of motion in segmentation

Reference Books

1. R C Gonzalez , R E Woods, Digital Image Processing, 3rd Edition, Pearson Education.2008.
2. A K Jain, Fundamentals of Digital image Processing, Prentice Hall of India.1989.
3. K R Castleman, Digital Image Processing, Pearson Education.1996
4. Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons.1989.
5. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins,' Digital Image Processing using MATLAB', Pearson Education, Inc., 2004.

Digital Image Processing Lab

Practical: 60 Lectures

1. Write program to read and display digital image using MATLAB or SCILAB
   a. Become familiar with SCILAB/MATLAB Basic commands
   b. Read and display image in SCILAB/MATLAB
   c. Resize given image
   d. Convert given color image into gray-scale image
   e. Convert given color/gray-scale image into black & white image
   f. Draw image profile
   g. Separate color image in three R G & B planes
   h. Create color image using R, G and B three separate planes
   i. Flow control and LOOP in SCILAB
   j. Write given 2-D data in image file

2. To write and execute image processing programs using point processing method
   a. Obtain Negative image
   b. Obtain Flip image
   c. Thresholding
   d. Contrast stretching

3. To write and execute programs for image arithmetic operations
   a. Addition of two images
   b. Subtract one image from other image
   c. Calculate mean value of image
   d. Different Brightness by changing mean value

4. To write and execute programs for image logical operations
   a. AND operation between two images
   b. OR operation between two images
   c. Calculate intersection of two images
   d. Water Marking using EX-OR operation
   e. NOT operation (Negative image)

5. To write a program for histogram calculation and equalization using

a. Standard MATLAB function
b. Program without using standard MATLAB functions
c. C Program

6. To write and execute program for geometric transformation of image
   a. Translation
   b. Scaling
   c. Rotation
   d. Shrinking
   e. Zooming

7. To understand various image noise models and to write programs for
   a. image restoration
   b. Remove Salt and Pepper Noise
   c. Minimize Gaussian noise
   d. Median filter and Weiner filter

8. Write and execute programs to remove noise using spatial filters
   a. Understand 1-D and 2-D convolution process
   b. Use 3x3 Mask for low pass filter and high pass filter

9. Write and execute programs for image frequency domain filtering
   a. Apply FFT on given image
   b. Perform low pass and high pass filtering in frequency domain
   c. Apply IFFT to reconstruct image

10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask

11. Write and execute program for image morphological operations erosion and dilation.

12. To write and execute program for wavelet transform on given image and perform inverse wavelet transform to reconstruct image.

V. Cryptographic Applications (DSE-III)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Understand and learn various public key as well as secret key cryptographic algorithms.
- Implement several cryptographic algorithms in the laboratory.
- Learn about different cyber security measures.

Theory: 60 Lectures

1. Introduction
Security Attacks, Computer Criminals, Security Services, Security Mechanisms. **7 Lectures**

2. Cryptography
Substitution ciphers, Transpositions Cipher, Confusion, diffusion, Symmetric, Asymmetric Encryption., DES Modes of DES, Uses of Encryption, Discrete Logarithm, Diffie-Hellman Problem, RSA algorithm, Hash function, key exchange, Digital Signatures, Digital Certificates. **25L**

3. Program Security
Secure programs, Non malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Control against program **10 Lectures**

4. Threats.
Protection in OS: Memory and Address Protection, Access control, File Protection, User Authentication. **9 Lectures**

5. Database Security
Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security. 9 **Lectures**

6. Security in Networks
Threats in Networks, Security Controls, firewalls, Intrusion detection systems, Secure e-mails

Recommended Books:

1. C. P. Pfleeger, S. L. Pfleeger; Security in Computing, Prentice Hall of India, 2006
2. B.A.Forouzan, Introduction to Cryptography and Network Security, Mc-Graw Hill
3. W. Stallings; Network Security Essentials: Applications and Standards, 4/E, 2010

Cryptographic Applications Lab
Lab Practical: 60 lectures

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig.
2. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations.
3. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these operations.
4. Design and implement a product cipher using substitution and transposition ciphers.
5. Perform encryption and decryption of affine cipher. Write a script for performing these operations.
6. Implement Diffie Hellman Key exchange algorithm
7. Implement RSA public key cryptosystem
8. Demonstrate sending of a protected word document.
9. Demonstrate sending of a digitally signed document.
10. Demonstrate sending of a protected worksheet.
11. Demonstrate use of steganography tools.

VI. Data Analytics (DSE-III)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Develop the ability to build and assess data-based models.
- Execute statistical analyses with professional statistical software
- Demonstrate skill in data management.

Theory: 60 Lectures

1. **Data Scientist's Tool Box**: Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio. **12L**

2. **R Programming Basics**: Overview of R, R data types and objects, reading and writing data, Control structures, functions, scoping rules, dates and times, Loop functions, debugging tools, Simulation, code profiling **12L**

3. **Getting and Cleaning Data**: Obtaining data from the web, from APIs, from database sand from colleagues in various formats. basics of data cleaning and making data ―tidy‖. **12L**

4. **Exploratory Data Analysis**: Essential exploratory techniques for summarizing data, applied before formal modeling commences, eliminating or sharpening potential hypotheses about the world that can be addressed by the data, common multivariate statistical techniques used to visualize high-dimensional data. **12L**

5. **Reproducible Research** : Concepts and tools behind reporting modern data analyses in are producible manner, To write a document using R markdown, integrate live R code into a literate statistical program, compile R markdown documents using knitr and related tools, and organize a data analysis so that it is reproducible and accessible to others. **12L**

Reference Books

1. Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontiline" by Schroff/O'Reilly, 2013.

2. Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking" by O'Reilly, 2013.

3. John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.

4. Ian Ayres, "Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart" Ist Edition by Bantam, 2007.

5. Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1st Edition, by Wiley, 2013.

6. Matthew A. Russel, "Mining the Social Web: Data mining Facebook, Twitter, Linkedln, Goole+, GitHub, and More", Second Edition, by O'Reilly Media, 2013.

Data Analytics Lab

Practical: 60 Lectures

1. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
2. Write a program that prints _Hello World‘ to the screen.
3. Write a program that prints a multiplication table for numbers up to 12.
4. Write a function that returns the largest element in a list.
5. Write a function that computes the running total of a list. 6. Write a function that tests whether a string is a palindrome.
7. Implement linear search.
8. Implement binary search.
9. Implement matrices addition , subtraction and Multiplication
10. Fifteen students were enrolled in a course. There ages were: 20 20 20 20 20 21 21 21 22 22 22 22 23 23 23
    i. Find the median age of all students under 22 years
    ii. Find the median age of all students
    iii. Find the mean age of all students
    iv. Find the modal age for all students
    v. Two more students enter the class. The age of both students is 23. What is now mean, mode and median?
11. Following table gives a frequency distribution of systolic blood pressure. Compute all the measures of dispersion.

Midpoint	95.5	105.5	115.5	125.5	135.5	145.5	155.5	165.5	175.5
Number	5	8	22	27	17	9	5	5	2

12. Obtain probability distribution of , where X is number of spots showing when a six-sided symmetric die (i.e. all six faces of the die are equally likely) is rolled. Simulate random samples of sizes 40, 70 and 100 respectively and verify the frequency interpretation of probability.
13. Make visual representations of data using the base, lattice, and ggplot2 plotting systems in R, apply basic principles of data graphics to create rich analytic graphics from available datasets.
14. Use Git / Github software to create Github account. Also, create a repo using Github.

VII. Project Work/Dissertation (DSE-IV)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Apply fundamental and disciplinary concepts and methods in ways appropriate to their principal areas of study.
- Demonstrate skill and knowledge of current information and technological tools and techniques specific to the professional field of study.

Practical Based:

- This option is to be offered only in 6^{th} Semester.
- The students will be allowed to work on any project based on the concepts studied in core/elective or skill based elective courses.
- The group size should be maximum of three (03) students.
- Each group will be assigned a teacher as a supervisor who will handle both their theory as well lab classes.
- A maximum of Four (04) projects would be assigned to one teacher.
- Theory classes will cover project management techniques.

SKILL ENHANCEMENT COURSES (Credit 02 Each)

(Any Two to be selected: SEC 1 - SEC 2)

1 Programming in Python (SEC-1)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- The course is designed to provide Basic knowledge of Python.
- Students will be able to acquire programming skills in core Python.
- Students will be able to acquire Object Oriented Skills in Python.
- Students will be able to solve problems requiring the writing of well-documented programs in the Python language, including use of the logical constructs of that language.

Theory: 15 Lectures

1. Planning the Computer Program: (2L)

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation.

2. Techniques of Problem Solving: (2L)

Flow charting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming. Overview of Programming
Structure of a Python Program, Elements of Python

3. **Introduction to Python: (7L)**

Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment/ Decrement Operator

4. Creating Python Programs: (4L)

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

Reference Books

a. T. Budd, Exploring Python, TMH, 1st Ed, 2011
b. Python Tutorial/Documentation www.python.or2015
c. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012
d. http://docs.python.org/3/tutorial/index.html
e. http://interactivepython.org/courselib/static/pythonds
f. http://www.ibiblio.org/g2swap/byteofpython/read/

Software Lab Based on Python:

Section: A (Simple programs)

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon user's choice.
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:
   Grade A: Percentage >=80 Grade B: Percentage>=70 and <80 Grade C: Percentage>=60 and <70 Grade D: Percentage>=40 and <60 Grade E: Percentage<40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to find sum of the following series for n terms: 1 – 2/2! + 3/3! ----------n/n!
7. WAP to calculate the sum and product of two compatible matrices.

Section: B (Visual Python):

All the programs should be written using user defined functions, wherever possible.

1. Write a menu-driven program to create mathematical 3D objects
   I. curve
   II. sphere
   III. cone
   IV. arrow
   V. ring
   VI. Cylinder.
2. WAP to read n integers and display them as a histogram.
3. WAP to display sine, cosine, polynomial and exponential curves.
4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user.
5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula m=60/(t+2), where t is the time in hours. Sketch a graph for t vs. m, where t>=0.
6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows:
   $P(t) = (15000(1+t))/(15+ e)$
   Where the time t is measured in hours. WAP to determine the size of the population at given time t and plot a graph for P vs t for the specified time interval.
7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion:
   I. velocity wrt time (v=u+at)
   II. distance wrt time (s=u*t+0.5*a*t*t)
   III. distance wrt velocity (s=(v*v-u*u)/2*a)
8. WAP to show a ball bouncing between 2 walls. (Optional)

2. UNIX/ LINUX Programming (SEC-1)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Effectively use the UNIX/Linux system to accomplish typical personal, office, technical, and software development tasks.
- Monitor system performance and network activities.
- Scripts and programs will demonstrate simple effective user interfaces.

Theory: 15 Lectures

Introduction (4L)

What is linux/unix Operating systems

a. Difference between linux/unix and other operating systems
b. Features and Architecture
c. Various Distributions available in the market
d. Installation, Booting and shutdown process
e. System Processes (an Overview)
f. Internal and External Commands
g. Creation of Partitions in O/S
h. Processes and its creation Phases- Fork, Exec, wait

User Management and File Systems (5L)

a. Types of Users, Creating users, Granting rights
b. User management commands
c. File quota and various file systems available
d. File System Management and Layout, File permissions
e. Login process, Managing Disk Quotas
f. Links (hard links, symbolic links)

Shell introduction and shell scripting (6L)

a. What is shell and various type of shell, Various editors present in linux
b. Different modes of operation in vi editor
c. What is shell script, Writing and executing the shell script
d. Shell variable (user defined and system variables)
e. System calls, Using system calls
f. Pipes and Filters
g. Decision making in Shell Scripts (If else, switch), Loops in shell
h. Functions
i. Utility programs (cut, paste, join, tr , uniq utilities)
j. Pattern matching utility (grep)

Reference Books:

1. Sumitabha, Das, Unix Concepts And Applications, Tata McGraw-Hill Education, 2006
2. Michael Jang RHCSA/ RHCE Red Hat Linux Certification: Exams (Ex200 & Ex300) (Certification Press), 2011
3. Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education, 2nd Edition ,2010
4. W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking API, Vol. 1, 3rd Edition,2014

Software Lab Based on UNIX/ Linux:

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify ―cal‖ command to display calendars of the specified months.
3. Write a shell script to modify ―cal‖ command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message – ―Entered login name is invalid‖.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of ―who‖ command along with the total number of users .
7. Write a shell script to display the multiplication table any number,
8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number.
10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
11. Write a shell script to find the LCD (least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the binomial coefficient C (n, x).
15. Write a shell script to find the permutation P (n, x).
16. Write a shell script to find the greatest number among the three numbers.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.

3. HTML Programming (SEC-2)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Use the HTML programming language.
- Resolves written HTML codes.
- Runs the page he/she has designed using HTML codes.
- Design simple web site and pages through HTML programming.

Theory: 15 Lectures

UNIT I: **Introduction** **(1L)**

UNIT II: **Basics** **(2L)**

Head, Body, Colors, Attributes Lists, Ordered, Unordered

UNIT III: **Links** **(3L)**

Introduction to Relative and Absolute Links Link Attributes Using the ID Attribute to Link within a Document

UNIT IV: **Images** **(2L)**

Putting an Image on a Page Using Images as Links Putting an Image in the Background

UNIT V: **Tables** **(4L)**

Creating a Table, Table Headers, Captions, Spanning Multiple Columns Styling Table

UNIT VI: **Forms** **(3L)**

Basic Input and Attributes Other Kind of Inputs Styling Forms with CSS Where to Go from Here

HTML Programming: LAB

Software Lab Based on HTML:

A. 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. Paragraph
   VII. Line Break
   VIII. Horizontal Rule
   IX. Pre tag

B. Create an HTML document which consists of:
   I. Ordered List
   II. Unordered List
   III. Nested List of Images

C. Create a table having students' Roll, Name and Grade

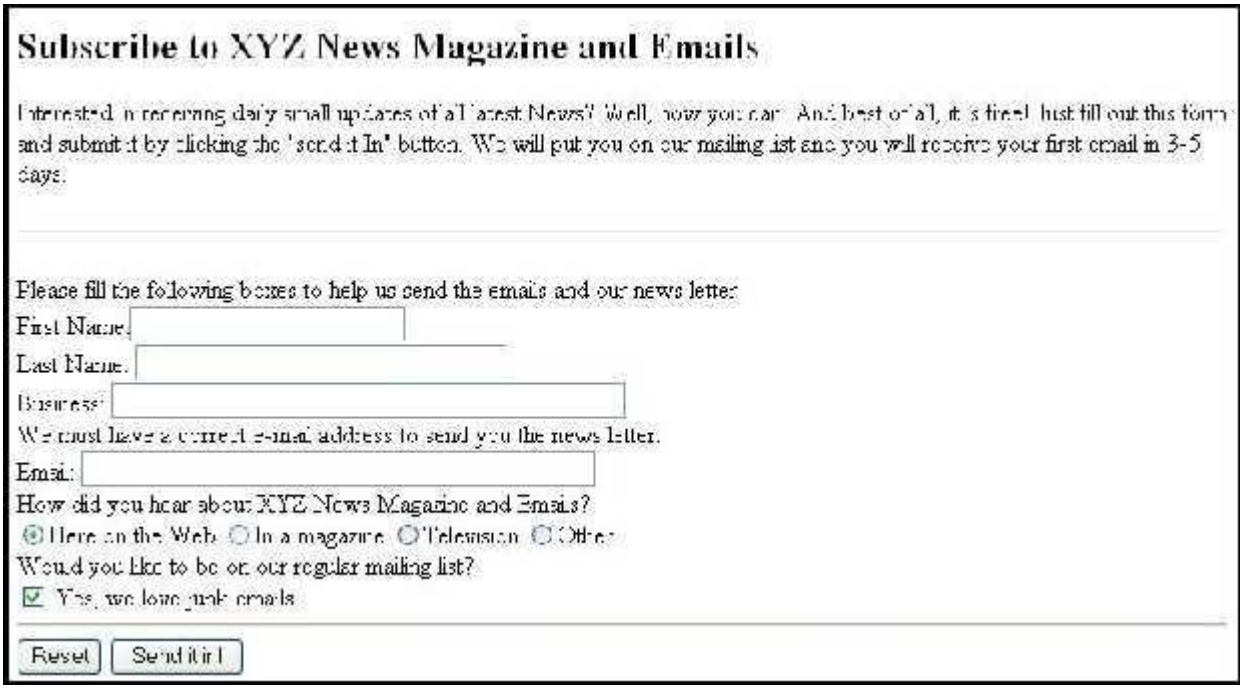
D. Create a Table with an inserted image

E. Create a form using HTML which has the following types of controls:
   - Text Box
   - Option/radio buttons
   - Check boxes
   - Reset and Submit buttons

F. Create HTML documents (having multiple frames) in the following three formats:

<table>
<tr><td colspan="2">Frame 1</td></tr>
<tr><td>Frame 2</td><td>Frame 3</td></tr>
</table>

<table>
<tr><td>Frame 1</td></tr>
<tr><td>Frame 2</td></tr>
</table>



4 PHP Programming (SEC-2)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Learn the basics and history of PHP and how to write your own PHP documents.
- Write regular expressions including modifiers, operators, and metacharacters.
- Create PHP programs that use various PHP library functions, manipulate files and directories.
- Write PHP scripts to handle HTML forms.

Theory: 15 Lectures

1. Introduction to PHP

Evaluation of Php, Basic Syntax, Defining variable and constant, Php Data type, Operator and Expression.

2. Decisions and loop

Making Decisions, Doing Repetitive task with looping, Mixing Decisions and looping with Html.

3. Function

What is a function, Define a function, Call by value and Call by reference, Recursive function, String Creating and accessing, String Searching & Replacing String, Formatting String, String Related Library function

4. Array

Anatomy of an Array, Creating index based and Associative array Accessing array, Element Looping with Index based array, Looping with associative array using each () and for each(), Some useful Library function.

5. Handling Html Form with Php

Capturing Form, Data Dealing with Multi-value filed, and Generating File uploaded form, redirecting a form after submission.

6. Working with file and Directories

Understanding file & directory, Opening and closing, a file, Coping, renaming and deleting a file, working with directories, Creating and deleting folder, File Uploading & Downloading.

7. Database Connectivity with MySql

Introduction to RDBMS, Connection with MySql Database, Performing basic database operation (DML) (Insert, Delete, Update, Select), Setting query parameter, Executing query Join (Cross joins, Inner joins, Outer Joins, Self joins.)

8. Exception Handling

Understanding Exception and error, Try, catch, throw. Error tracking and debugging

Reference Books:

1. Learning PHP, MySQL, books by ‘ O’ reily Press
2. The complete reference PHP, Steven Holzner
3. PHP beginer’s practical guide, Pratiyush Guleria, BPB Publication

PHP Programming: LAB

Software Lab Based on PHP:

1. write a factorial program using for loop in PHP
2. Write a program to check student grade based on marks
3. Write a program to show day of a week using switch
4. Write a program to calculate to sum of digits of a number
5. Write a PHP program to display 1 to 100 using PHP
6. Write a PHP program that check whether a given string is palindrome or not using function.
7. Write a PHP program to check if a person is eligible to vote using function
8. Write a program to print two variables in single echo
9. Write a PHP program to reverse the string
10. Write a PHP script to find maximum number out of three given numbers.

General Elective Papers (GE) (Minor – Computer Science) (any four: GE1 - GE 4) for other Departments/Disciplines: (Credit: 06 each)

1 Introduction to Programming: (GE-1)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Become familiar with basic structure and operations of a computer.
- Get familiar with different approaches for solving a problem through computer.
- Get a simple basic knowledge about Computer Organization.
- Develop and run simple C programs.

Theory: 60 Lectures

1. Computer Fundamentals: Introduction to Computers: Characteristics of Computers, Uses of Computers, Types and Generations of Computers **(3L)**

2. Basic Computer Organization - Units of a computer, CPU, ALU, memory hierarchy, registers, I/O Devices. **(4L)**

3. Planning the Computer Program: Concept of problem solving, Problem definition, Program Design, debugging, types of programming errors, documentation **(3L)**

4. Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming. **(4L)**

5. Introduction to C **(3L)**

History of C, Overview of Procedural Programming and Object-Orientation Programming, Using main() function, Compiling and Executing Simple Programs in C.

6. Data Types, Variables, Constants, Operators and Basic I/O **(5L)**

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()), Using Basic Header Files (stdio.h, conio.hetc).

7. Expressions, Conditional Statements and Iterative Statements **(6L)**

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)

8. Functions and Arrays **(11L)**

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

9. Derived Data Types (Structures and Unions) (4L)

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

10. Pointers in C (8L)

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with 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

11. Memory Allocation in C (4L)

Differentiating between static and dynamic memory allocation, use of malloc, calloc and free functions, storage of variables in static and dynamic memory allocation

12. File I/O, Pre-processor Directives (5L)

Opening and closing a file, Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives, Macros.

Reference Books:

1. P. K. Sinha & Priti Sinha , "Computer Fundamentals", BPB Publications, 2007.
2. Dr. Anita Goel, Computer Fundamentals, Pearson Education, 2010.
3. T. Budd, Exploring Python, TMH, 1st Ed, 2011
4. Python Tutorial/Documentation www.python.or2010
5. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012
6. http://docs.python.org/3/tutorial/index.html
7. http://interactivepython.org/courselib/static/pythonds
8. http://www.ibiblio.org/g2swap/byteofpython/read/

COMPUTER SCIENCE LAB: Introduction to Programming
Lab Practical:

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() / calloc() functions or new operator.

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

i) Show address of each character in string
j) Concatenate two strings without using strcat function.
k) Concatenate two strings using strcat function.
l) Compare two strings
m) Calculate length of the string (use pointers)

n) Convert all lowercase characters to uppercase
o) Convert all uppercase characters to lowercase
p) Calculate number of vowels

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 a Matrix using 2D array. Write a menu-driven program to perform following Matrix operations (2-D array implementation):
a) Sum b) Difference c) Product d) Transpose

2 Introduction to Database Systems (GE-2)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Know about different database management techniques.
- They can create, delete, and update the databases in practical classes.
- Through database management system they can work with any real life database.

Theory: 60 lectures

1. **Database:** (**14 L)**

Introduction to database, relational data model, DBMS architecture, data independence, DBA, database users, end users, front end tools

2. E-R Modeling: (14 L)

Entity types, entity set, attribute and key, relationships, relation types, E- R diagrams, database design using ER diagrams

3. Relational Data Model: (14 L)

Relational model concepts, relational constraints, primaryand foreign key, normalization: 1NF, 2NF, 3NF

4. Structured Query Language: (18 L)

SQL queries, create a database table, createrelationships between database tables, modify and manage tables, queries, forms, reports, modify, filter and view data.

Reference Books :

1. P. Rob, C. Coronel, Database System Concepts by, Cengage Learning India, 2008
2. R. Elmsasri,S. Navathe Fundamentals of Database Systems, Pearson Education, Fifth Edition, 2007
3. MySQL : Reference Manual

Introduction to Database Systems Lab
Practical: 60 lectures

1) Create a database having two tables with the specified fields, to computerize a library system of a Delhi University College.
LibraryBooks (Accession number, Title, Author, Department, PurchaseDate, Price) IssuedBooks (Accession number, Borrower)

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
b) Delete the record of book titled ―Database System Concepts‖.
c) Change the Department of the book titled ―Discrete Maths‖ to ―CS‖.
d) List all books that belong to ―CS‖ department.
e) List all books that belong to ―CS‖ department and are written by author ―Navathe‖.
f) List all computer (Department=‖CS‖) that have been issued.
g) List all books which have a price less than 500 or purchased between ―01/01/1999‖ and ―01/01/2004‖.

Create a database having in your college.

2) **Personal information about Student (College roll number, Name of student, Date of birth, Address, Marks(rounded off to whole number) in percentage at 10 + 2, Phone number) Paper Details (Paper code, Name of the Paper)**
Student's Academic and Attendance details (College roll number, Paper code, Attendance, Marks in home examination).

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
b) Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper 2.
c) List all students who live in ―Delhi‖ and have marks greater than 60 in paper 1.
d) Find the total attendance and total marks obtained by each student.
e) List the name of student who has got the highest marks in paper 2.

3) Create the following tables and answer the queries given below: **Customer (CustID, email, Name, Phone, ReferrerID) Bicycle (BicycleID, DatePurchased, Color, CustID, ModelNo) BicycleModel (ModelNo, Manufacturer, Style) Service (StartDate, BicycleID, EndDate)**

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
b) List all the customers who have the bicycles manufactured by manufacturer ―Honda‖.
c) List the bicycles purchased by the customers who have been referred by customer ―C1‖.
d) List the manufacturer of red colored bicycles.
e) List the models of the bicycles given for service.

4) Create the following tables, enter at least 5 records in each table and answer the queries given below.

EMPLOYEE (Person_Name, Street, City)
WORKS (Person_Name, Company_Name,
Salary) COMPANY (Company_Name, City)
MANAGES (Person_Name, Manager_Name)

a) Identify primary and foreign keys.
b) Alter table employee, add a column ―email‖ of type varchar(20).
c) Find the name of all managers who work for both Samba Bank and NCB Bank.
d) Find the names, street address and cities of residence and salary of all employees who work for ―Samba Bank‖ and earn more than $10,000.
e) Find the names of all employees who live in the same city as the company for which they work.
f) Find the highest salary, lowest salary and average salary paid by each company.
g) Find the sum of salary and number of employees in each company.
h) Find the name of the company that pays highest salary.

5) Create the following tables, enter at least 5 records in each table and answer the queries given below.

Suppliers (SNo, Sname, Status,
SCity) Parts (PNo, Pname,
Colour, Weight, City) Project
(JNo, Jname, Jcity) Shipment
(Sno, Pno, Jno, Qunatity)

a) Identify primary and foreign keys.
b) Get supplier numbers for suppliers in Paris with status>20.
c) Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.
d) Get suppliers names for suppliers who do not supply part P2.
e) For each shipment get full shipment details, including total shipment weights.
f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.
g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.
h) Get the names of cities that store more than five red parts.
i) Get full details of parts supplied by a supplier in London.
j) Get part numbers for part supplied by a supplier in London to a project in London.
k) Get the total number of project supplied by a supplier (say, S1).
l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).

3 Computer Networks and Internet Technologies (GE-3)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Acquire the computer networking knowledge as well as the existing connectivity technologies.
- Know about the OSI and TCP/IP model of communications.
- The practical course enables them to design web applications using HTML and Java script.

Theory: 60 Lectures

Computer Networks: Introduction to computer networks, data communication and components of data communication, data transmission mode, data communication measurement, LAN, MAN, WAN, wireless LAN, internet, intranet, extranet **6L**

Network Models: Client/ server network and Peer-to-peer network, OSI, TCP/IP, layers and functionalities. **8L**

Transmission Media: Introduction, Guided Media: Twisted pair, Coaxial cable, Optical fiber. Unguided media: Microwave, Radio frequency propagation, Satellite. **4L**

LAN Topologies: Ring, bus, star, mesh and tree topologies. **2L**

Network Devices: NIC, repeaters, hub, bridge, switch, gateway and router. **2L**

Internet Terms: Web page, Home page, website, internet browsers, URL, Hypertext, ISP, Web server, download and upload, online and offline. **2L**

Internet Applications: www, telnet, ftp, e-mail, social networks, search engines, Video Conferencing, e-Commerce, m-Commerce, VOIP, blogs. **6L**

Introduction to Web Design: Introduction to hypertext markup language (html) Document type definition, creating web pages, lists, hyperlinks, tables, web forms, inserting images, frames, hosting options and domain name registration. Customized **16L**

Features: Cascading style sheet (css) for text formatting and other manipulations.

JavaScript Fundamentals: Data types and variables, functions, methods and events, controlling program flow, JavaScript object model, built-in objects and operators. **14L**

Reference Books:

1. Andrew S. Tanenbaum, David J. Wetherall Computer Networks (5th Edition),PHI, 2010
2. B. A. Forouzan, Data Communication and Networking , TMH,2003.
3. D.R. Brooks, An Introduction to HTML and Javascript for Scientists and Engineers, Springer W. Willard,2009
4. HTML A Beginner's Guide, Tata McGraw-Hill Education, 2009.
5. J. A. Ramalho, Learn Advanced HTML 4.0 with DHTML, BPB Publications, 2007

Computer Networks and Internet Technologies Lab Practical: 60 lectures

Practical exercises based on concepts listed in theory using HTML.

1. Create HTML document with following formatting – Bold, Italics, Underline, Colors, Headings, Title, Font and Font Width, Background, Paragraph, Line Brakes, Horizontal Line, Blinking text as well as marquee text.

2. Create HTML document with Ordered and Unordered lists, Inserting Images, Internal and External linking
3. Create Form with Input Type, Select and Text Area in HTML.

4. Create an HTML containing Roll No., student's name and Grades in a tabular form.

5. Create an HTML document (having two frames) which will appear as follows:

About Department 1 Department 2 Department 3	This frame would show the contents According to the link clicked on the left frame

6. Create an HTML document containing horizontal frames as follows:

Department Names (could be along with Logos)

Contents according to the Link clicked

7. Create a website of 6 – 7 pages with different effects as mentioned in above problems.

8. Create HTML documents (having multiple frames) in the following three formats:

Frame1	
Frame2	Frame3

9. Create a form using HTML which has the following types of controls:
    I. Text Box
    II. Option/radio buttons
    III. Check boxes
    IV. Reset and Submit buttons

List of Practical's using Java Script :

Create event driven program for following:

1. Print a table of numbers from 5 to 15 and their squares and cubes using alert. 8. Print the largest of three numbers.
2. Find the factorial of a number n.
3. Enter a list of positive numbers terminated by Zero. Find the sum and average of these numbers.
4. 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.
5. Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

4 Programming in Python (GE-4)

Course Learning Outcomes: After successful completion of the Course a student will be able to:

- Learn the basic features of Python programming.
- Acquire programming skills in core Python.
- Develop simple programs of Python and run them in their practical classes.
- In practical classes students can learn the concepts by writing programs.

Theory: 60 lectures

Computer Fundamentals: Introduction to Computers: Characteristics of Computers, Uses of computers, Types and generations of Computers. **(4L)**

Basic Computer Organization - Units of a computer, CPU, ALU, memory hierarchy, registers, I/O devices. **(6L)**

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. **(4L)**

Techniques of Problem Solving: Flowchart, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming. **(4L)**

Overview of Programming: Structure of a Python program, Elements of Python **(6L)**

Introduction to Python: Python Interpreter, Using Python as calculator, Python shell, Indentation. Atoms, Identifiers and keywords, Literals, Strings, Operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment, Operator, Ternary operator, Bit wise operator, Increment or Decrement operator). **(8L)**

Creating Python Programs: Input and Output Statements, Control statements (Looping-while Loop, for Loop , Loop Control, Conditional Statement- if...else, Difference between break, continue and pass). **(14L)**

Structures: Numbers, Strings, Lists, Tuples, Dictionary, Date & Time, Modules, Defining Functions, Exit function, default arguments. **(14L)**

References :

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. How to think like a computer scientist : learning with Python / Allen Downey, Jeffrey Elkner, Chris Meyers. 1st Edition – Freely available online. 2012
3. http://docs.python.org/3/tutorial/index.html
4. http://interactivepython.org/courselib/static/pythonds

Programming in Python

Lab Practical: 60 lectures

1. Using for loop, print a table of Celsius/Fahrenheit equivalences. Let c be the Celsius temperatures ranging from 0 to 100, for each value of c, print the corresponding Fahrenheit temperature.
2. Using while loop, produce a table of sins, cosines and tangents. Make a variable x in range from 0 to 10 in steps of 0.2. For each value of x, print the value of sin(x), cos(x) and tan(x).
3. Write a program that reads an integer value and prints ―leap year‖ or ―not a leap year‖.
4. Write a program that takes a positive integer n and then produces n lines of output shown as follows.
   For example enter a size: 5
   *
   **
   ***
   ****
   *****
5. Write a function that takes an integer _n‘ as input and calculates the value of 1 + 1/1! + 1/2! + 1/3! + … + 1/n
6. Write a function that takes an integer input and calculates the factorial of that number.
7. Write a function that takes a string input and checks if it‘s a palindrome or not.
8. Write a list function to convert a string into a list, as in list (_abc‘) gives [a, b, c].
9. Write a program to generate Fibonacci series.
10. Write a program to check whether the input number is even or odd.
11. Write a program to compare three numbers and print the largest one.
12. Write a program to print factors of a given number.
13. Write a method to calculate GCD of two numbers.