The University of Burdwan

B.Sc. Honours in Computer Science

CBCS w.e.f. 2017 – 2018

SEMESTER – I

Course code	Course Title	Credits	No. of Hours/Week		
			L	T	P
CC-1	Programming Fundamentals using C / C++	4-0-2=6	4	0	4
CC-2	Computer System Architecture	4-0-2=6	4	0	4
GE-1	*From other discipline*	6			
AECC-1	Environmental studies	4			
		22			

SEMESTER – II

Course code	Course title	Credits	No. of Hours/Week		
			L	T	P
CC-3	Programming in Java	4-0-2=6	4	0	4
CC-4	Discrete Structure	5-1-0=6	5	1	0
GE-2	*From other discipline*	6			
AECC-2	ENG/HIND/MIL	2			
		20			

SEMESTER – III

Course code	Course title	Credits	No. of Hours/Week		
			L	T	P
CC-5	Data Structure	4-0-2=6	4	0	4
CC-6	Operating Systems	4-0-2=6	4	0	4
CC-7	Computer Networks	4-0-2=6	4	0	4
GE-3	*From other discipline*	6			
SEC-1	Programming in Python **OR** R Programming	1-0-1=2	1	0	2
		26			

SEMESTER – IV

Course code	Course title	Credit	No of Hours		
			L	T	P
CC-8	Design and Analysis of Algorithms	4-0-2=6	4	0	4
CC-9	Software Engineering	4-0-2=6	4	0	4
CC-10	Database Management System	4-0-2=6	4	0	4
GE-4	*From other discipline*	6			
SEC-2	PHP Programming **OR** UNIX / Linux Programming	1-0-1=2	1	0	2
		26			

SEMESTER – V

Course code	Course title	Credit	No. of Hours/Week		
			L	T	P
CC-11	Internet Technologies	4-0-2=6	4	0	4
CC-12	Theory of Computation	5-1-0=6	5	1	0
DSE-1	Microprocessors **OR** Digital Image Processing	4-0-2=6	4	0	4
DSE-2	Numerical Methods **OR** System Programming	4-0-2=6	4	0	4
		24			

SEMESTER – VI

Course code	Course title	Credit	No. of Hours/Week		
			L	T	P
CC-13	Artificial Intelligence	4-0-2=6	4	0	4
CC-14	Computer Graphics	4-0-2=6	4	0	4
DSE-3	Combinatorial Optimization **OR** Soft Computing	4-0-2=6	4	0	4
DSE-4	Data Mining **OR** Project Work / Dissertation	4-0-2=6	4	0	4
		24			

Core Course (CC) Papers: (14 papers, each carrying a credit of 6 each (4 Theory + 2 Practical))

Paper-Code	Paper-Name
CC-1	Programming Fundamentals using C / C++
CC-2	Computer System Architecture
CC-3	Programming in Java
CC-4	Discrete Structures
CC-5	Data Structure
CC-6	Operating Systems
CC-7	Computer Networks
CC-8	Design and Analysis of Algorithms
CC-9	Software Engineering
CC-10	Database Management System
CC-11	Internet Technologies
CC-12	Theory of Computation
CC-13	Artificial Intelligence
CC-14	Computer Graphics

Discipline Specific Elective (DSE) Papers: (4 papers, each carrying a credit of 6 each (4 Theory + 2 Practical))

DSE 1

Microprocessor

OR

Digital Image Processing

DSE 2

Numerical Methods

OR

System Programming

DSE 3

Combinatorial optimization

OR

Soft Computing

DSE4

Data Mining

OR

Project Work / Dissertation

SEC= Skill Enhancement Course (2 Papers, each carrying a credit of 2 (1 Theory + 1 Practical)):

SEC 1

Programming in Python

OR

R Programming

SEC 2

PHP Programming

OR

UNIX / Linux Programming

General Electives for other disciplines (GE 1 to GE 4) to be offered by Computer Science Department

GE 1:

Problem Solving using Computer

GE 2:

Database Management Systems

GE 3:

Operating Systems

GE 4:

Computer System Architecture

SEMESTER - I

CC-1 (Theory): Programming Fundamentals using C / C++

Theory: 60 Lectures **Credit: 4**

1. Introduction to C and C++ (3 L)

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

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

Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putcharetc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc).

3. Expressions, Conditional Statements and Iterative Statements (5 L)

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 L)

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) Twodimensional 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 L)

Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.

6. Pointers and References in C++ (7 L)

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Problems with 11 Pointers, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values

7. Memory Allocation in C++ (3 L)

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

8. File I/O, Preprocessor Directives (4 L)

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

9. Using Classes in C++ (7 L)

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

10. Overview of Function Overloading and Operator Overloading (5 L)

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

11. Inheritance, Polymorphism and Exception Handling (8 L)

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

CC-1 (Practical): Programming Fundamentals using C/C++

Practical: 60 Lectures **Credit: 2**

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

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

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

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

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

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

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

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

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

a) Calculate surface Area
b) Calculate Volume
c) Increment, Overload ++ operator (both prefix & postfix)
d) Decrement, Overload -- operator (both prefix & postfix)
e) Overload operator == (to check equality of two boxes), as a friend function
f) Overload Assignment operator
g) Check if it is a Cube or cuboid

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

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

26. Write a program to retrieve the student information from file created in previous question and print it in following format: Roll No. Name Marks

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

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

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

Reference Books

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

CC-2 (Theory): Computer System Architecture

Theory: 60 Lectures **Credit: 4**

1. Introduction **(8 L)**

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 L)**

Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison, multiplication and division algorithms for integers

3. Basic Computer Organization and Design **(13 L)**

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 L)**

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 L)**

Cache memory, Associative memory, mapping.

6. Input-Output Organization **(8 L)**

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

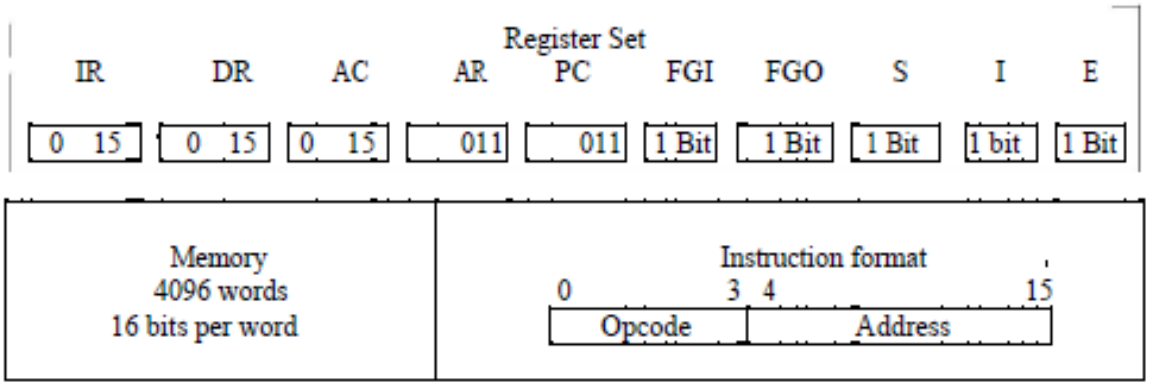
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India,2009

4. M.M. Mano , Digital Design, Pearson Education Asia,2013 5. Carl Hamacher, Computer Organization, Fifth edition, McGrawHill, 2012.

CC-2 (Practical): Computer System Architecture

Practical: 60 Lectures **Credit: 2**

1. Create a machine based on the following architecture:



Basic Computer Instructions

Memory Reference			Register Reference		Input-Output		
Symbol	Hex		Symbol	Hex	Symbol	Hex	
AND	0xxx		CLA	E800	INP	F80 0	
ADD	2xxx		CLE	E400	OUT	F40 0	
LDA	4xxx	Direct	CMA	E200	SKI	F20 0	
STA	6xxx	Addressing	CME	E100	SKO	F10 0	
BUN	8xxx		CIR	E080	ION	F08 0	Optional
BSA	Axxx		CIL	E040	IOF	F04 0	
ISZ	Cxxx		INC	E020			
AND_I	1xxx		SPA	E010			
ADD_I	3xxx		SNA	E008			
LDA_I	5xxx	Indirect	SZA	E004			
STA_I	7xxx	Addressing	SZE	E002			
BUN_I	9xxx		HLT	E001			
BSA_I	Bxxx						
ISZ_I	Dxxx						

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

(except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

3. Create a Fetch routine of the instruction cycle.

4. Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

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

Initialize the contents of AC to $(A937)_{16}$, that of PC to $(022)_{16}$ and E to 1.

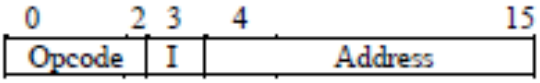
5. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

a. ADD	f. BSA
b. AND	g. ISZ
c. LDA	
d. STA	
e. BUN	

6. Simulate the machine for the memory-reference instructions referred in above question with I= 1 and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

7. Modify the machine created in Practical 1 according to the following instruction format:

Instruction format



   a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. There are only two addressing modes, I = 0 (direct addressing) and I = 1 (indirect addressing).

   b. Create a new register I of 1 bit.

   c. Create two new microinstructions as follows :

      i. Check the opcode of instruction to determine type of instruction (Memory Reference/Register Reference/Input-Output) and then jump accordingly.

      ii. Check the I bit to determine the addressing mode and then jump accordingly.

For other disciplines

GE-1: Problem Solving using Computer

Theory: 60 Lectures Credit: 4

Computer Fundamentals:

Introduction to Computers: Characteristics of Computers, Uses of computers, Types and generations of Computers. (3L)

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

Planning the Computer Program:

Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. (3L)

Techniques of Problem Solving:

Flowcharting, 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 (4L)

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). (10L)

Structures:

Numbers, Strings, Lists, Tuples, Dictionary, Date & Time, Modules, Defining Functions, Exit function, default arguments. (10L)

Introduction to Advanced Python:

Objects and Classes, Inheritance, Regular Expressions, Event Driven Programming, GUI Programming. (14L)

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.or 2010
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/

Software Lab using Python: Credit : 2

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)

SEMESTER - II

CC-3 (Theory): Programming in Java

Theory: 60 lectures **Credit: 4**

1. Introduction to Java **(4 L)**

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

2. Arrays, Strings and I/O **(8 L)**

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

3. Object-Oriented Programming Overview **(14 L)**

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

4. Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata **(14 L)**

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

5. Exception Handling, Threading, Networking and Database Connectivity **(10 L)**

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.

6. Applets and Event Handling **(10 L)**

Java Applets:Introduction to Applets, Writing Java Applets, Working with Graphics,
Incorporating Images & Sounds. Event Handling Mechanisms,Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls.

CC-3 (Practical): Programming in Java

Practical: 60 Lectures **Credit: 2**

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .lenth in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard

7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument

8. Write a program that show working of different functions of String and StringBufferclasss like setCharAt(, setLength(), append(), insert(), concat()and equals().

9. Write a program to create a ―distance‖ class with methods where distance is computed in terms of feet and inches, how to create objects of a class and to see the use of this pointer

10. Modify the ―distance‖ class by creating constructor for assigning values (feet and inches) to the distance object. Create another object and assign second object as reference variable to another object reference variable. Further create a third object which is a clone of the first object.

11. Write a program to show that during function overloading, if no matching argument is found, then java will apply automatic type conversions(from lower to higher data type)

12. Write a program to show the difference between public and private access specifiers. The program should also show that primitive data types are passed by value and objects are passed by reference and to learn use of final keyword

13. Write a program to show the use of static functions and to pass variable length arguments in a function.

14. Write a program to demonstrate the concept of boxing and unboxing.

15. Create a multi-file program where in one file a string message is taken as input from the user and the function to display the message on the screen is given in another file (make use of Scanner package in this program).

16. Write a program to create a multilevel package and also creates a reusable class to generate Fibonacci series, where the function to generate fibonacii series is given in a different file belonging to the same package.

17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages

18. Write a program ―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.

19. Write a program to show the use of nested try statements that emphasizes the sequence of checking for catch handler statements.

20. Write a program to create your own exception types to handle situation specific to your application (Hint: Define a subclass of Exception which itself is a subclass of Throwable).

21. Write a program to demonstrate priorities among multiple threads.

22. Write a program to demonstrate multithread communication by implementing synchronization among threads (Hint: you can implement a simple producer and consumer problem).

23. Write a program 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.
24. Write a program to get the URL/location of code (i.e. java code) and document(i.e. html file).

25. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
26. Write a program to demonstrate different keyboard handling events.
27. Write a program to generate a window without an applet window using main() function.

28. Write a program to demonstrate the use of push buttons.

Reference Books

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

CC-4 (Theory): Discrete Structures

Theory: 60 lectures **Credit: 5**

1. **Introduction:** 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. (15 L)
2. **Growth of Functions:** Asymptotic Notations, Summation formulas and properties, Bounding Summations, approximation by Integrals. (8 L)
3. **Recurrences:** Recurrence Relations, Generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem. (10 L)
4. **Graph Theory:** Basic Terminology, Models and Types, multi-graphs 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. (15 L)
5. **Propositional Logic:** Logical Connectives, Well-formed Formulas, Tautologies, Equivalences, Inference Theory. (12 L)

CC-4 (Tutorial): Discrete Structures

Tutorial: 10 lectures **Credit: 1**

Tutorial based on concepts listed in 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

For other disciplines

GE-2: Database Management Systems

Theory: 60 Lectures Credit: 4

Introduction to Database Management Systems: Characteristics of database approach, data models, DBMS architecture and data independence. (10L)
Entity Relationship and Enhanced ER Modeling: Entity types, relationships, SQL- 99: Schema Definition, constraints, and object modeling. (15L)
Relational Data Model: Basic concepts, relational constraints, relational algebra, SQL queries. (15L)
Database design: ER and EER to relational mapping, functional dependencies, normal forms up to third normal form. (20 L)

Software Lab based on Database Management Systems Credit: 2

Note: MS Access/MySQL may be used.
The following concepts must be introduced to the students:

DDL Commands
• Create table, alter table, drop table

DML Commands
• Select , update, delete, insert statements
• Condition specification using Boolean and comparison operators (and, or,

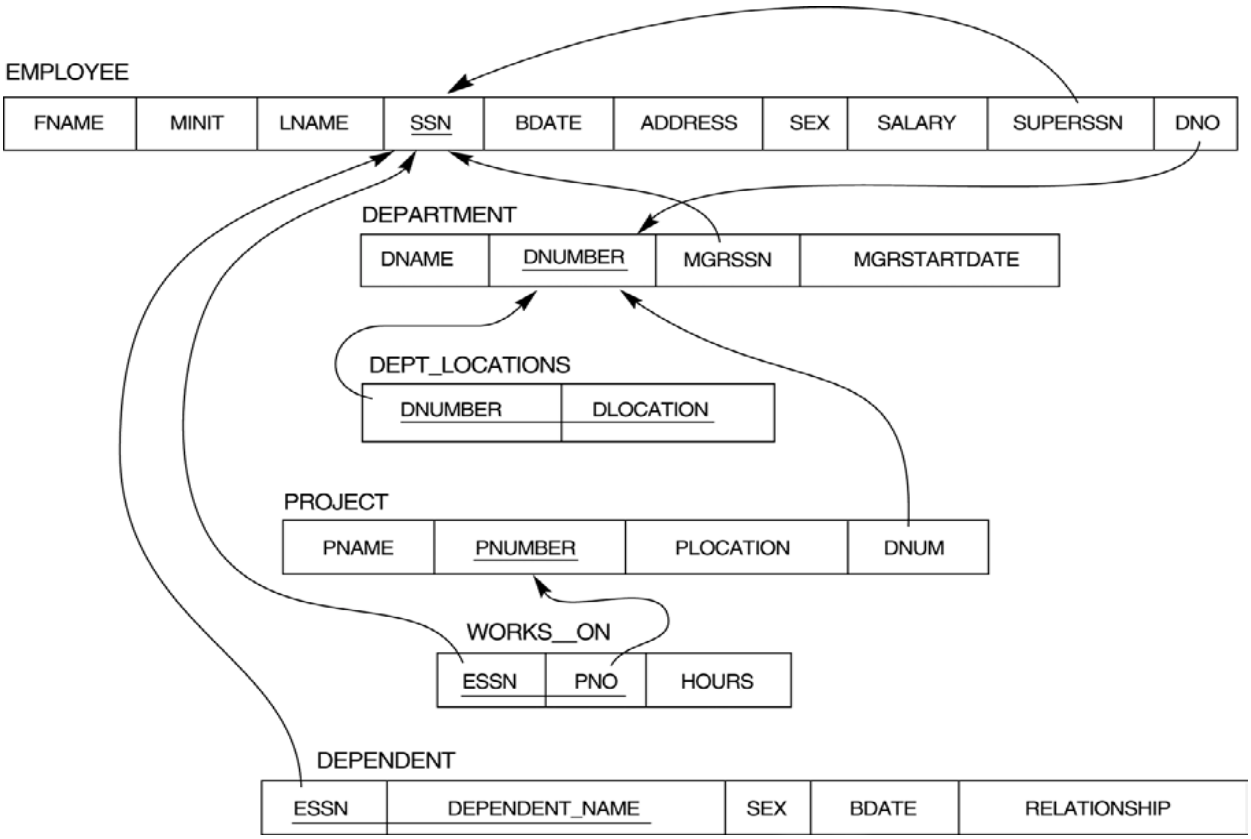
not,=,<>,>,<,>=,<=)
• Arithmetic operators and aggregate functions(Count, sum, avg, Min, Max)
• Multiple table queries (join on different and same tables)
• Nested select statements
• Set manipulation using (any, in, contains, all, not in, not contains, exists, not

exists, union, intersect, minus, etc.)
• Categorization using group by……..having
• Arranging using order by

Reference Books:

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

Relational Database Schema - COMPANY



Questions to be performed on above schema

1. Create tables with relevant foreign key constraints
2. Populate the tables with data
3. Perform the following queries on the database :

1. Display all the details of all employees working in the company.
2. Display ssn, lname, fname, address of employees who work in department no 7.

3. Retrieve the birthdate and address of the employee whose name is 'Franklin T. Wong'
4. Retrieve the name and salary of every employee
5. Retrieve all distinct salary values
6. Retrieve all employee names whose address is in ‘Bellaire’
7. Retrieve all employees who were born during the 1950s
8. Retrieve all employees in department 5 whose salary is between 50,000 and 60,000(inclusive)
9. Retrieve the names of all employees who do not have supervisors
10. Retrieve SSN and department name for all employees
11. Retrieve the name and address of all employees who work for the 'Research' department
12. For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate.
13. For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.
14. Retrieve all combinations of Employee Name and Department Name

15. Make a list of all project numbers for projects that involve an employee whose last name is 'Narayan' either as a worker or as a manager of the department that controls the project.
16. Increase the salary of all employees working on the 'ProductX' project by 15%. Retrieve employee name and increased salary of these employees.
17. Retrieve a list of employees and the project name each works in, ordered by the employee's department, and within each department ordered alphabetically by employee first name.
18. Select the names of employees whose salary does not match with salary of any employee in department 10.
19. Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee.
20. Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford.
21. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.
22. Find the sum of the salaries and number of employees of all employees of the 'Marketing' department, as well as the maximum salary, the minimum salary, and the average salary in this department.
23. Select the names of employees whose salary is greater than the average salary of all employees in department 10.
24. For each department, retrieve the department number, the number of employees in the department, and their average salary.
25. For each project, retrieve the project number, the project name, and the number of employees who work on that project.
26. Change the location and controlling department number for all projects having more than 5 employees to 'Bellaire' and 6 respectively.
27. For each department having more than 10 employees, retrieve the department no, no of employees drawing more than 40,000 as salary.
28. Insert a record in Project table which violates referential integrity constraint with respect to Department number. Now remove the violation by making necessary insertion in the Department table.
29. Delete all dependents of employee whose ssn is '123456789'.
30. Delete an employee from Employee table with ssn = '12345'(make sure that this employee has some dependents, is working on some project, is a manager of some department and is supervising some employees). Check and display the cascading effect on Dependent and Works on table. In Department table MGRSSN should be set to default value and in Employee table SUPERSSN should be set to NULL
31. Perform a query using alter command to drop/add field and a constraint in Employee table.

SEMESTER-III

CC-5 (Theory): Data Structures

Theory: 60 Lectures **Credit: 4**

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

2. **Stacks** **(5 L)**
   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 L)**
   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 L)**
   Array and Linked representation of Queue, De-queue, Priority Queues, Queue using Stack

5. **Recursion** **(5 L)**
   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 L)**
   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);

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

8. **Hashing** **(5 L)**
   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, YedidyahLangsam, "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; 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. ___

CC—5 (Practical): Data Structures Lab

Practical: 60 Lectures **Credit: 2**

All programs should be developed in C/ C++/Java/Python

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

CC-6: Operating Systems

Theory: 60 Lectures **Credit: 4**

1. Introduction (10 L)

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 L)

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

3. Process Management **(20** L)

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 L)

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

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

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

6. Protection and Security (4 L)

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.

CC-6: Operating Systems Lab

Practical: 60 Lectures **Credit: 2**

All programs should be developed in C/ C++/Java/Python

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.

CC-7: Computer Networks

Theory: 60 Lectures **Credit: 4**

1. Introduction to Computer Networks (8 L)
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 L)
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 (twisted-pair, coaxial-cable, fibre-optic cable), satellite communications.

3. Network Switching Techniques and Access mechanisms (10 L)
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 L)
Error detection and error correction techniques (CRC, Hamming-codes); 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 L)
ALOHA & CSMA protocols; CDMA; Ethernet LANs; connecting LAN and back-bone networks- repeaters, hubs, switches, bridges, router and gateways;

6. Networks Layer Functions and Protocols (6 L)
Routing; static and dynamic routing algorithms; network layer protocol of Internet- IP protocol, Internet control protocols.

7. Transport Layer Functions and Protocols (6 L)
Transport services- error and flow control, Connection establishment and release- three way handshake, TCP and UDP.

8. Overview of Application layer protocol (5 L)
Overview of DNS protocol; overview of WWW & HTTP protocol.

Reference Books

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

CC-7 (Practical): Computer Networks Lab

Practical: 60 Lectures **Credit: 2**

All programs should be developed in C/ C++ / Java / Python

1. Simulate Cyclic Redundancy Check (CRC) error detection algorithm for noisy channel.
2. Simulate Hamming-code based error detection & correction algorithm for noisy channel.
3. Simulate and implement stop and wait protocol for noisy channel.
4. Simulate and implement go back N sliding window protocol.
5. Simulate and implement selective repeat sliding window protocol.
6. Simulate and implement MST construction (Prim's, Kruskal's) for Ethernet
7. Simulate and implement the various routing algorithms (RIP, Distance-Vector routing, Dijkstra's, Bellman-Ford, Floyd-Warshall, Flooding).

SEC-1

SEC-1: Programming in Python (1 Th. + 2 Labs)

Theory: 15 Lectures **Credit: 1**

Planning the Computer Program: Concept of problem solving, Problem definition, Program design, Debugging, Types of errors in programming, Documentation. (2L)

Techniques of Problem Solving: Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming. (2 L)

Overview of Programming: Structure of a Python Program, Elements of Python (3 L)

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). (4L)

Creating Python Programs : Input and Output Statements, Control statements(Branching, Looping, Conditional Statement, Exit function, Difference between break, continue and pass.), Defining Functions, default arguments. (4L)

Reference Books

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. Python Tutorial/Documentation www.python.or 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python , Freely available online.2012
4. http://docs.python.org/3/tutorial/index.html
5. http://interactivepython.org/courselib/static/pythonds
6. http://www.ibiblio.org/g2swap/byteofpython/read/

Software Lab Based on Python:

Practical: 30 Lectures **Credit: 2**

Section: A (Simple programs)

1. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users 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 paramters 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)

SEC-1: R-Programming (1 Th.+2 Labs)

Theory: 15 Lectures **Credit: 1**

Introduction: Overview and History of R, Getting Help, Data Types, Subsetting, Vectorized Operations, Reading and Writing Data. (5L)

Control Structures, Functions, lapply, tapply, split, mapply, apply, Coding Standards. (5L)

Scoping Rules, Debugging Tools, Simulation, R Profiler. (5L)

Reference Books

1. William N. Venables and David M. Smith, An Introduction to R. 2nd Edition. Network Theory Limited.2009
2.Norman Matloff, The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011

Software Lab Based on R Programming

Practical: 30 Lectures **Credit: 2**

1. Write a program that prints ‗Hello World' to the screen.
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n
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 the following sorting algorithms: Selection sort, Insertion sort, Bubble Sort
8. Implement linear search.
9. Implement binary search.
10. Implement matrices addition, subtraction and Multiplication

GE-3: Operating Systems

Theory: 60 Lectures Credit: 4

Introduction: System Software, Resource Abstraction, OS strategies. (2L)

Types of operating systems - Multiprogramming, Batch, Time Sharing, Single user and Multiuser, Process Control & Real Time Systems. (2L)

Operating System Organization: Factors in operating system design, basic OS functions, implementation consideration; process modes, methods of requesting system services – system calls and system programs. (10L)

Process Management : System view of the process and resources, initiating the OS, process address space, process abstraction, resource abstraction, process hierarchy, Thread model (15L)

Scheduling: Scheduling Mechanisms, Strategy selection, non-pre-emptive and pre-emptive strategies. (12L)

Memory Management: Mapping address space to memory space, memory allocation strategies, fixed partition, variable partition, paging, virtual memory (12L)

Shell introduction and Shell Scripting **(7L)**

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

Books Recommended:

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.

Software Lab based on Operating Systems Credit: 2

Note: Following exercises can be performed using Linux or Unix

1. Usage of following commands:
   ls, pwd, tty, cat, who, who am I, rm, mkdir, rmdir, touch, cd.
2. Usage of following commands:
   cal, cat(append), cat(concatenate), mv, cp, man, date.
3. Usage of following commands:
   chmod, grep, tput (clear, highlight), bc.
4. Write a shell script to check if the number entered at the command line is prime or not.
5. Write a shell script to modify "cal" command to display calendars of the specified months.
6. Write a shell script to modify "cal" command to display calendars of the specified range of months.
7. Write a shell script to accept a login name. If not a valid login name display message – "Entered login name is invalid".
8. Write a shell script to display date in the mm/dd/yy format.
9. Write a shell script to display on the screen sorted output of "who" command along with the total number of users.
10. Write a shell script to display the multiplication table any number,
11. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
12. Write a shell script to find the sum of digits of a given number.
13. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
14. Write a shell script to find the LCD(least common divisor) of two numbers.
15. Write a shell script to perform the tasks of basic calculator.
16. Write a shell script to find the power of a given number.
17. Write a shell script to find the factorial of a given number.
18. Write a shell script to check whether the number is Armstrong or not.
19. Write a shell script to check whether the file have all the permissions or not.
20. Program to show the pyramid of special character "*".

SEMESTER – IV

CC-8 (Theory): Design and Analysis of Algorithms

Theory: 60 Lectures **Credit: 4**

1. Introduction (5 L)

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

2. Algorithm Design Techniques (8 L)

Iterative techniques, Divide and Conquer (Karatsuba integer multiplication, Strassen's matrix multiplication, etc.), Dynamic Programming (Convex hull, DFT & FFT), Greedy Algorithms (Knapsack, Denomination of notes/coins), Scheduling.

3. Sorting and Searching Techniques (17 L)

Elementary sorting techniques– 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: Decision Trees (2 L)

5. Balanced Trees: AVL Trees, Red-Black Trees (10 L)

6. Advanced Analysis Technique: Amortized analysis (5 L)

7. Graphs (5 L)

Graph Algorithms–Breadth First Search, Depth First Search and its Applications, Minimum Spanning Trees (Prim, Kruskal), Shortest Paths (Dijkstra, Bellman-Ford, Floyd-Warshall).

8. **String Processing:** String Matching, Edit Distance, Document distance, Rabin-Karp, KMP, Boyer-Moore Techniques (5 L)

9. **Cryptographic algorithms and Overview of complexity Classes:** Basic cryptographic algorithms like DSA, RSA; P, NP, NP-Hard, NP-Complete (3 L)

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

CC-8 (Practical): Design and Analysis of Algorithms Lab

Practical: 60 Lectures **Credit: 2**

All programs should be developed in C/ C++/Java/Python

1. i. Implement Insertion Sort (The program should report the number of comparisons)
   ii. Implement Merge Sort(The program should report the number of comparisons)
2. Implement Heap Sort(The program should report the number of comparisons)
3. Implement Randomized Quick sort (The program should report the number of comparisons)
4. Implement Radix Sort
5. Create a Red-Black Tree and perform following operations on it:
   i. Insert a node
   ii. Delete a node

iii. Search for a number & also report the color of the node containing this number.

6. Write a program to determine the LCS of two given sequences
7. Implement Breadth-First Search in a graph
8. Implement Depth-First Search in a graph
9. Write a program to determine the minimum spanning tree of a graph

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

CC-9: Software Engineering

Theory: 60 Lectures **Credit: 4**

1. Introduction (8 L)

The Evolving Role of Software, Software Characteristics, Changing Nature ofSoftware, Software Engineering as a Layered Technology, Software Process Framework, Framework and Umbrella Activities, Process Models, Capability Maturity Model Integration (CMMI).

2. Requirement Analysis (10 L)

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

3. Software Project Management (8L)

Estimation in Project Planning Process, Project Scheduling.

4. Risk Management (8 L)

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

5.Quality Management (8 L)

Quality Concepts, Software Quality Assurance, Software Reviews,Metrics for Process and Projects.

6. Design Engineering (10 L)

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

7. Testing Strategies & Tactics (8 L)

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

CC-9 (Practical): Software Engineering Lab

Practical: 60 Lectures **Credit: 2**

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:
   - •Basis Path Testing

Sample Projects:

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

CC-10: Database Management Systems

Theory: 60 Lectures

Credit: 4

1.Introduction (6 L)

Characteristics of database approach, data models, database system architecture and data independence.

2.Entity Relationship (ER) Modeling (8 L)

Entity types, relationships, constraints.

3.Relation data model (20 L)

Relational model concepts, relational constraints, relational algebra, SQL queries

4. Database design (15 L)

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

5. Transaction Processing (3 L)

ACID properties, concurrency control & recovery

6. File Structure and Indexing (8 L)

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

Books Recommended:

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

CC-10 (Practical): Database Management Systems

Lab Practical: 60 Lectures **Credit: 2**

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.

SEC-2

PHP Programming (1 +2 Lab)

Theory: 15 Lectures **Credit: 1**

Introduction to PHP: (3L)

- PHP introduction, inventions and versions, important tools and software requirements (like Web Server, Database, Editors etc.)
- PHP with other technologies, scope of PHP
- Basic Syntax, PHP variables and constants
- Types of data in PHP , Expressions, scopes of a variable (local, global)
- PHP Operators : Arithmetic, Assignment, Relational , Logical operators, Bitwise , ternary and MOD operator.
- PHP operator Precedence and associativity

Handling HTML form with PHP: (2L)

Capturing Form Data

- GET and POST form methods
- Dealing with multi value fields
- Redirecting a form after submission

PHP conditional events and Loops: (3L)

PHP IF Else conditional statements (Nested IF and Else)

- Switch case, while ,For and Do While Loop
- Goto , Break ,Continue and exit

PHP Functions: (3L)

Function, Need of Function , declaration and calling of a function

- PHP Function with arguments, Default Arguments in Function
- Function argument with call by value, call by reference
- Scope of Function Global and Local

String Manipulation and Regular Expression: (2L)

- Creating and accessing String , Searching & Replacing String
- Formatting, joining and splitting String , String Related Library functions
- Use and advantage of regular expression over inbuilt function
- Use of preg_match(), preg_replace(), preg_split() functions in regular expression

Array: (2L)

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

Reference Books:

1. Steven Holzner, "PHP: The Complete Reference Paperback", McGraw Hill Education (India), 2007.

2. Timothy Boronczyk, Martin E. Psinas, "PHP and MYSQL (Create-Modify-Reuse)", Wiley India Private Limited, 2008.

3. Robin Nixon, "Learning PHP, MySQL, JavaScript, CSS & HTML5", 3rd Edition Paperback, O'reilly, 2014.

4. Luke Welling, Laura Thompson, PHP and MySQL Web Development", 4th Edition, Addition Paperback, Addison-Wesley Professsional,2008.

5. David Sklar, Adam Trachtenberg, "PHP Cookbook: Solutions & Examples for PHP Programmers", 2014.

Software Lab Based on PHP:

Practical: 30 Lectures **Credit: 2**

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.
5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? (A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.
   Sample string : 'The quick " " brown fox' Expected Output : Thequick""brownfox
9. Write a PHP script that finds out the sum of first n odd numbers.
10. Create a login page having user name and password. On clicking submit, a welcome message should be displayed if the user is already registered (i.e.name is present in the database) otherwise error message should be displayed.
11. Write a PHP script that checks if a string contains another string.
12. Create a simple 'birthday countdown' script, the script will count the number of days between current day and birth day.
13. Create a script to construct the following pattern, using nested for loop.

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

14. Write a simple PHP program to check that emails are valid.
15. WAP to print first n even numbers.
16. $color = array('white', 'green', 'red")
    Write a PHP script which will display the colors in the following way : Output : white, green, red,
    - green
    - red
    - white
17. Using switch case and dropdown list display a —Hello‖ message depending on the language selected in drop down list.
18. Write a PHP program to print Fibonacci series using recursion.
19. Write a PHP script to replace the first 'the' of the following string with 'That'.

    Sample : 'the quick brown fox jumps over the lazy dog.'
    Expected Result : That quick brown fox jumps over the lazy dog.

SEC-2: UNIX / Linux Programming (1+2 Labs)

Theory: 15 Lectures **Credit: 1**

Introduction (5 L)

- What is linux/unix Operating systems
- Difference between linux/unix and other operating systems
- Features and Architecture
- Various Distributions available in the market
- Installation, Booting and shutdown process
- System processes (an overview)
- External and internal commands
- Creation of partitions in OS
- Processes and its creation phases – Fork, Exec, wait

User Management and the File System (5L)

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

Shell introduction and Shell Scripting (6L)

- What is shell and various type of shell, Various editors present in linux
- Different modes of operation in vi editor
- What is shell script, Writing and executing the shell script
- Shell variable (user defined and system variables)
- System calls, Using system calls
- Pipes and Filters
- Decision making in Shell Scripts (If else, switch), Loops in shell
- Functions
- Utility programs (cut, paste, join, tr , uniq utilities)
- 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 Linux:

Practical: 30 Lectures **Credit: 2**

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.

GE- 4: Computer System Architecture

Theory: 60 Lectures Credit: 4

Introduction: Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexors, registers, counters and memory units. (12L)

Data Representation and basic Computer Arithmetic: Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison. (8L)

Basic Computer Organization and Design: Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt. (18L)

Central Processing Unit: Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. (10L)

Programming the Basic Computer: Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming. (8L)

Input-output Organization: Peripheral devices, I/O interface, Modes of data transfer, direct memory access. (4L)

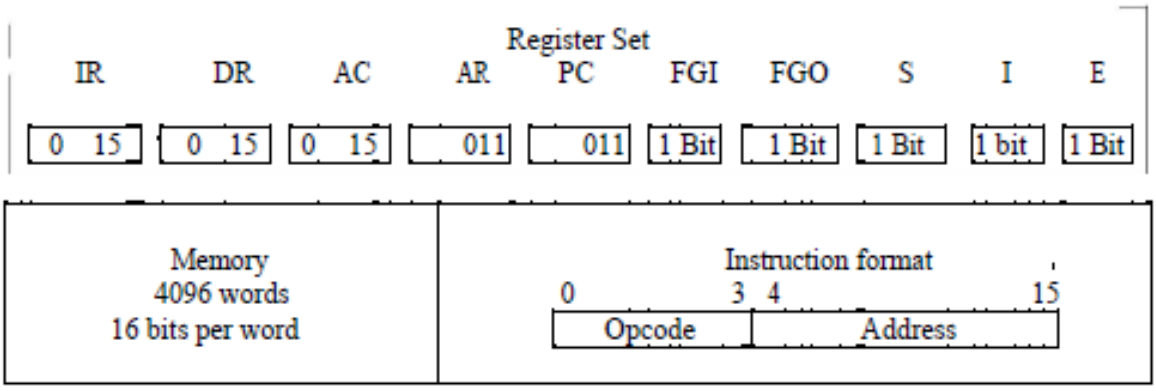
Books Recommended:

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. Digital Design, M.M. Mano, Pearson Education Asia, 1979

Computer System Architecture Lab

Practical: 60 Lab Periods

1. Create a machine based on the following architecture:



Basic Computer Instructions

Memory Reference			Register Reference		Input-Output		
Symbol	Hex		Symbol	Hex	Symbol	Hex	
AND	0xxx		CLA	E800	INP	F80 0	
ADD	2xxx		CLE	E400	OUT	F40 0	
LDA	4xxx	Direct	CMA	E200	SKI	F20 0	
STA	6xxx	Addressing	CME	E100	SKO	F10 0	
BUN	8xxx		CIR	E080	ION	F08 0	Optional
BSA	Axxx		CIL	E040	IOF	F04 0	
ISZ	Cxxx		INC	E020			
AND_I	1xxx		SPA	E010			
ADD_I	3xxx		SNA	E008			
LDA_I	5xxx	Indirect	SZA	E004			
STA_I	7xxx	Addressing	SZE	E002			
BUN_I	9xxx		HLT	E001			
BSA_I	Bxxx						
ISZ_I	Dxxx						

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

ii) Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.

iii) Create a Fetch routine of the instruction cycle.

iv) Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:

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

Initialize the contents of AC to $(A937)_{16}$, that of PC to $(022)_{16}$ and E to 1.

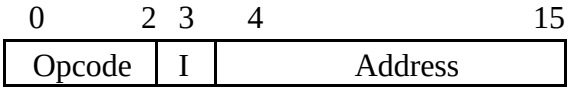
5. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

a. ADD	f. BSA
b. AND	g. ISZ
c. LDA	
d. STA	
e. BUN	

6. Simulate the machine for the memory-reference instructions referred in above question with I= 1 and address part = 082. The instruction to be stored at address 026 in RAM. Initialize the memory word at address 082 with the value 298. Initialize the memory word at address 298 with operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.

7. Modify the machine created in Practical 1 according to the following instruction format:

Instruction format



   a. The instruction format contains a 3-bit opcode, a 1-bit addressing mode and a 12-bit address. There are only two addressing modes, I = 0 (direct addressing) and I = 1 (indirect addressing).
   b. Create a new register I of 1 bit.
   c. Create two new microinstructions as follows :
      i. Check the opcode of instruction to determine type of instruction (Memory Reference/Register Reference/Input-Output) and then jump accordingly.
      ii. Check the I bit to determine the addressing mode and then jump accordingly.

SEMESTER – V

CC-11 (Theory): Internet Technologies

Theory: 60 Lectures **Credit: 4**

Java: Use of Objects, Array and Array List class **(5 L)**

JavaScript: Data types, operators, functions, control structures, events and event handling. **(15 L)**

JDBC **(10 L)**

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

JSP **(20 L)**

Introduction to JavaServer 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.

Java Beans **(10 L)**

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

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.

CC-11 (Practical): Internet Technologies Lab

Practical: 60 Lectures **Credit: 2**

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.

Read n numbers. Count the number of negative numbers, positive numbers and zeros in the list.

CC-12 (Theory): Theory of Computation

Theory: 60 Lectures **Credit: 5**

1. Languages (8 L)

Alphabets, string, language, Basic Operations on language, Concatenation, KleeneStar

2. Finite Automata and Regular Languages (20 L)

Regular Expressions, Transition Graphs, Deterministics 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, Moore & Mealy Machines.

3. Context free languages (17 L)

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 Macines and Models of Computations (15 L)

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

Recommended Books:

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

CC-12 (Tutorial): Theory of Computation

Tutorial: 15 Lectures **Credit: 1**

Tutorial based on theory classes.

DSE 1

Discipline Specific Elective Papers COMPUTER SCIENCE: (Credit: 06 each)

Microprocessors

Theory: 60 Lectures **Credit: 4**

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. (20 L)

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

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

Recommended Books:

1. Ramesh Gaonkar: Microprocessor Architecture, Programming and Application with the 8085
2. Barry B. Brey : The Intel Microprocessors : Architecture, Programming and Interfacing. Pearson Education, Sixth Edition,2009.
3. Walter A Triebel, Avtar Singh; The 8088 and 8086 Microprocessors Programming, Interfacing, Software, Hardware, and Applications. PHI, Fourth Edition 2005.

Microprocessor Lab

Practical: 60 Lectures **Credit :2**

All programs should be developed for the Intel 8085 Microprocessor.

ASSEMBLY LANGUAGE PROGRAMMING

1. Write a program for 32-bit binary division and multiplication
2. Write a program for 32-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 BCD conversion
6. Write a program for BCD to binary conversion

OR

Digital Image Processing

Theory: 60 Lectures **Credit: 4**

1. Introduction **(6 L)**

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

2. Spatial Domain Filtering **(7 L)**

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

3. Filtering in the Frequency domain **(8 L)**

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 L)**

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 L)**

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 L)**

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 L)**

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 L)**

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.
6. William N. Venables and David M. Smith, An Introduction to R. 2nd Edition. Network Theory Limited.2009
7. Norman Matloff, The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011

Digital Image Processing Lab

Practical: 60 Lectures **Credit: 2**

All programs should be developed using C++ / Java / R / MATLAB / SCILAB / Python

1. Write program to do the following.
   a. Become familiar with Basic commands.
   b. Read and display image.
   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 Loops
   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 and equalization.
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 1D and 2D 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 for edge detection using different edge detection masks.
11. Write and execute program for image morphological operations erosion and dilation.

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

DSE-2:

Numerical Methods

Theory: 60 Lectures **Credit: 4**

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

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

Iterative methods: Jacobi and Gauss-Seidel iterative methods

Interpolation: Lagrange‘s form and Newton‘s form 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

Numerical integration: Trapezoid rule, Simpson‘s rule (only method), Newton−Cotes open formulas

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

Reference 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)

[4] William N. Venables and David M. Smith, An Introduction to R. 2nd Edition. Network Theory Limited.2009

[5] Norman Matloff, The Art of R Programming - A Tour of Statistical Software Design, No Starch Press.2011

Numerical Methods Lab

Practical: 60 lectures

All programs should be developed using C / C++ / Java / R / MATLAB / SCILAB / Python

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.

OR

Systems Programming

Theory: 60 lectures **Credit: 4**

Assemblers & Loaders, Linkers: (10 L)
One pass and two pass assembler, design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking.

Introduction: (2 L)
Overview of compilation, Phases of a compiler

Lexical Analysis: (6 L)
Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lex.

Parsing: (10 L)
Bottom up parsing- LR parser (SLR, LALR, CLR), yacc.

Intermediate representations (10 L)
Three address code generation & representations , syntax directed translation, translation of types, control
Statements

Storage organization: (5 L)
Activation records, stack allocation

Code Generation: (5 L)
Object code generation

Reference Books

1. Santanu Chattopadhyaya, *Systems Programming*, PHI, 2011.
2. Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, *Compilers: Principles, Techniques, and Tools*, 2nd edition, Prentice Hall, 2006.
3. D. M. Dhamdhere, *Systems Programming*, Tata McGraw Hill, 2011.
4. Leland Beck, D. Manjula, *System Software: An Introduction to System Programming*, 3rd edition, Pearson Education, 2008.
5. Grune D, Van Reeuwijk . K, Bal H. E, Jacobs C J H, Langendoen K, *Modern Compiler Design*, 2nd edition, Springer, 2012

Systems Programming Lab

Practical: 60 lectures **Credit: 2**

All programs should be developed in C/ C++/Java/Python

[1] To implement an assembler for a hypothetical language.
[2] To get familiar with lex: write a program to recognize numbers, identifiers.
[3] To get familiar with yacc: write a desk calculator.

SEMESTER – VI

CC-13 (Theory): Artificial Intelligence

Theory: 60 Lectures **Credit: 4**

1. Introduction (6 L)

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 L)

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 L)

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs.
Programming in Logic (PROLOG) / LISP

4. Dealing with Uncertainty and Inconsistencies (8 L)

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

5. Understanding Natural Languages (6 L)

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.

CC-13 (Practical): Artificial Intelligence Lab

Practical: 60 Lectures **Credit: 2**

All programs should be developed using Prolog or LISP

1. Write a prolog/lisp program to calculate the sum of two numbers.
2. Write a prolog/lisp program to find the maximum of two numbers.
3. Write a prolog/lisp program to calculate the factorial of a given number.
4. Write a prolog/lisp program to calculate the nth Fibonacci number.
5. Write a prolog/lisp 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/lisp program to remove the Nth item from a list.
7. Write a Prolog/lisp 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/lisp program to implement append for two lists.
9. Write a Prolog/lisp program to implement palindrome(List).
10. Write a Prolog/lisp program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
11. Write a Prolog/lisp program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
12. Write a Prolog/lisp program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
13. Write a Prolog/lisp 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/lisp program to implement reverse(List,ReversedList) that reverses lists.
15. Write a Prolog/lisp 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/lisp program to implement GCD of two numbers.
17. Write a prolog/lisp program that implements Semantic Networks/Frame Structures.

CC-14 (Theory): Computer Graphics

Theory: 60 Lectures **Credit: 4**

---

1. **Introduction** (5 L)
   Basic elements of Computer graphics, Applications of Computer Graphics.
2. **Graphics Hardware** (8 L)
   Architecture of Raster and Random scan display devices, input/output devices.
3. **Fundamental Techniques in Graphics** (22 L)
   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 L)
   Representing curves & Surfaces.
5. **Visible Surface determination** (8 L)
   Hidden surface elimination.
6. **Surface rendering** (7 L)
   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.

CC-14 (Practical): Computer Graphics Lab

Practical: 60 Lectures **Credit: 2**

All programs should be developed using C / C++/ Java / Python

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.

DSE-3

Discipline Specific Elective Papers COMPUTER SCIENCE: (Credit: 06 each)

Combinatorial Optimization

Theory: 60 Lectures **Credit: 4**

1) **Algorithmic Perspective to Simplex Method** (10 L)

LP Formulation, Geometry of Linear Programming Problems, Theory of Simplex Algorithm, Geometric interpretation of Degeneracy, Avoiding cycles, Methods for obtaining initial Basic Feasible Solutions.

2) **Assignment and Transportation Problems**: (10 L)

Algorithm for Hungarian Method. Solution of the Assignment Problems using Hungarian Method. Algorithm for Vogal's Approximation Method(VAM). Solution of the Transportation Problems using VAM method.

3) **Integer Programming** (10 L)

Integrality gap, Branch and Bound algorithm, Cutting-plane algorithm, Applications of these algorithms on Travelling Salesman.

4) **Problem Primal-Dual Algorithms** (10 L)

Interpretation of Dual, Optimality conditions for primal and dual, primal-dual algorithms based on complementary slackness, Primal-dual algorithms for shortest path problem, vertex cover and set cover.

5) **Network and Flow Algorithms** (20 L)

Single Source Shortest path algorithms – Bellman Ford algorithm, all pair shortest path algorithms – Floyd Warshall algorithm. Linear Programming formulations of shortest path problem, network flows and bipartite matching. Cycle-cancelling. Totally uni-modular matrices integral polyhedra.

Books Recommended:

1. C.H.Papadimitriou and K.Steiglitz, Combinatorial Optimization: Algorithms and complexity, Prentice Hall of India, 2006
2. K.Lange, Optimization, Springer, 2004
3. Mokhtar S.Bazaraa, John J. Jarvis and Hanif D. Sherali, Linear Programming and Network Flows, John Wiley & Sons, 2004
4. H.A.Taha, Operations Research: An Introduction(8th edition), Prentice Hall, 2006
5. Vijay V. Vazirani, Approximation algorithms, Springer, 2002. Tutorial

Combinatorial Optimization Lab

Practical: 60 lectures **Credit: 2**

All programs should be developed using C/ C++/ Java / Python / R / SCILAB

Practical exercises based on concepts listed in theory.

OR

Soft Computing

Theory: 60 Lectures **Credit: 4**

Module-I (10 L)

Introduction: Introduction to soft computing; introduction to fuzzy sets and fuzzy logic systems; introduction to biological and artificial neural network; introduction to Genetic Algorithm.

Module-II: Fuzzy Logic (20 L)

Classical Sets and Fuzzy Sets and Fuzzy relations : Operations on Classical sets, properties of classical sets, Fuzzy set operations, properties of fuzzy sets, cardinality, operations, and properties of fuzzy relations.

Membership functions : Features of membership functions, standard forms and boundaries, different fuzzification methods.

Fuzzy to Crisp conversions: Lambda Cuts for fuzzy sets, fuzzy Relations, Defuzzification methods.

Classical Logic and Fuzzy Logic: Classical predicate logic, Fuzzy Logic, Approximate reasoning and Fuzzy Implication

Fuzzy Rule based Systems: Linguistic Hedges, Fuzzy Rule based system – Aggregation of fuzzy Rules, Fuzzy Inference System, Mamdani Fuzzy Models – Takagi-Sugeno Fuzzy Models.

Module-III: Neural Networks (20 L)

Introduction to Neural Networks: Advent of Modern Neuroscience, Classical AI and Neural Networks, Biological Neurons and Artificial neural network; model of artificial neuron.

Learning Methods : Hebbian, competitive, Boltzman.

Neural Network models: Perceptron, Adaline and Madaline networks; single layer network; Delta rule and back-propagation; and multi layer networks.

Competitive learning networks: Kohonen self organizing networks (SOM), Hebbian learning; Hopfield Networks, Bidirectional Associative Memory (BAM).

Module-IV: Genetic Algorithms (10 L)

Genetic Algorithms: Simple GA, crossover and mutation, Multi-objective Genetic Algorithm (MOGA).

Text Books:

1. Fuzzy logic with engineering applications, Timothy J. Ross, John Wiley and Sons. 2. S. Rajasekaran and G.A.V.Pai, "Neural Networks, Fuzzy Logic and Genetic Algorithms", PHI
3. Principles of Soft Computing , S N Sivanandam, S. Sumathi, John Wiley & Sons
4. Genetic Algorithms in search, Optimization & Machine Learning by David E. Goldberg
5. Neuro-Fuzzy and Soft computing, Jang, Sun, Mizutani, PHI
6. Neural Networks: A Classroom Approach,1/e by Kumar Satish, TMH,
7. Genetic Algorithms in search, Optimization & Machine Learning by David E. Goldberg, Pearson/PHI
8. A beginners approach to Soft Computing, Samir Roy & Udit Chakraborty,
9. Soft Computing: Fundamentals and Applications, by D. K. Pratihar.

Soft Computing Lab

Practical: 60 lectures **Credit: 2**

All programs should be developed in C++ / Java / Python / R

Practical exercises based on concepts listed in theory, such as:

1. Implementing an ANN to train a model
2. Implementing a Kohonen SOM
3. Implementing a Hopfield network to memorize and rectify patterns
4. Implementing BAM to memorize
5. Implementing Genetic Algorithms to solve an optimization problem
6. Applying ANN for pattern recognition tasks

DSE-4
Data Mining

Theory: 60 lectures **Credit: 4**

Overview: Predictive and descriptive data mining techniques, supervised and unsupervised learning techniques, process of knowledge discovery in databases, pre-processing methods

Data Mining Techniques: Association Rule Mining, Classification and regression techniques, Bayesian Classifiers, SVM, Ensemble methods, Bootstrapping, Nearest-Neighbour classification, clustering techniques: K-Means, K-Medians, DBSCAN, DENCLUE, BIRCH, CURE, Apriori algorithm, Outlier detection, PCA.

Other Topics: Scalability and data management issues in data mining algorithms, measures of interestingness

Books Recommended:

1. Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Introduction to Data Mining, Pearson Education.2005.
2. Richard Roiger, Michael Geatz, Data Mining: A Tutorial Based Primer, Pearson Education 2003.
3. G.K. Gupta, Introduction to Data Mining with Case Studies, PHI,2006.
4. Soman K P, Diwakar Shyam, Ajay V Insight Into Data Mining: Theory And Practice, , PHI, 2006
5. Data Mining Techniques: Arun K. Pujari.

Data Mining Lab

Practical: 60 lectures **Credit: 2**

Practical exercises based on concepts listed in theory by writing programs in C++ / Java / MATLAB / R / SCILAB / Python.

OR

Dissertation / Project work

This option is to be offered only in 6th Semester.
The students will be allowed to work on any project based on the concepts studied in core / elective or skill based elective courses.
Instead of submitting a project, students may alternatively submit a research-oriented dissertation following standard format/guidelines of undergraduate thesis/dissertation.
Each group will be assigned a teacher as a supervisor who will handle both their theory as well lab classes.
The group size should be maximum of four (04) students.
A maximum of Four (04) projects would be assigned to one teacher.