

Choice Based Credit System (CBCS)

UNIVERSITY OF DELHI

DEPARTMENT OF COMPUTER SCIENCE

**UNDERGRADUATE PROGRAMME
(Courses effective from Academic Year 2015-16)**



SYLLABUS OF COURSES TO BE OFFERED
Core Courses, Elective Courses & Ability Enhancement Courses

Disclaimer: The CBCS syllabus is uploaded as given by the Faculty concerned to the Academic Council. The same has been approved as it is by the Academic Council on 13.7.2015 and Executive Council on 14.7.2015. Any query may kindly be addressed to the concerned Faculty.

Undergraduate Programme Secretariat

Preamble

The University Grants Commission (UGC) has initiated several measures to bring equity, efficiency and excellence in the Higher Education System of country. The important measures taken to enhance academic standards and quality in higher education include innovation and improvements in curriculum, teaching-learning process, examination and evaluation systems, besides governance and other matters.

The UGC has formulated various regulations and guidelines from time to time to improve the higher education system and maintain minimum standards and quality across the Higher Educational Institutions (HEIs) in India. The academic reforms recommended by the UGC in the recent past have led to overall improvement in the higher education system. However, due to lot of diversity in the system of higher education, there are multiple approaches followed by universities towards examination, evaluation and grading system. While the HEIs must have the flexibility and freedom in designing the examination and evaluation methods that best fits the curriculum, syllabi and teaching-learning methods, there is a need to devise a sensible system for awarding the grades based on the performance of students. Presently the performance of the students is reported using the conventional system of marks secured in the examinations or grades or both. The conversion from marks to letter grades and the letter grades used vary widely across the HEIs in the country. This creates difficulty for the academia and the employers to understand and infer the performance of the students graduating from different universities and colleges based on grades.

The grading system is considered to be better than the conventional marks system and hence it has been followed in the top institutions in India and abroad. So it is desirable to introduce uniform grading system. This will facilitate student mobility across institutions within and across countries and also enable potential employers to assess the performance of students. To bring in the desired uniformity, in grading system and method for computing the cumulative grade point average (CGPA) based on the performance of students in the examinations, the UGC has formulated these guidelines.

CHOICE BASED CREDIT SYSTEM (CBCS):

The CBCS provides an opportunity for the students to choose courses from the prescribed courses comprising core, elective/minor or skill based courses. The courses can be evaluated following the grading system, which is considered to be better than the conventional marks system. Therefore, it is necessary to introduce uniform grading system in the entire higher education in India. This will benefit the students to move across institutions within India to begin with and across countries. The uniform grading system will also enable potential employers in assessing the performance of the candidates. In order to bring uniformity in evaluation system and computation of the Cumulative Grade Point Average (CGPA) based on student's performance in examinations, the UGC has formulated the guidelines to be followed.

Outline of Choice Based Credit System:

- 1. Core Course:** A course, which should compulsorily be studied by a candidate as a core requirement is termed as a Core course.
- 2. Elective Course:** Generally a course which can be chosen from a pool of courses and which may be very specific or specialized or advanced or supportive to the discipline/ subject of study or which provides an extended scope or which enables an exposure to some other discipline/subject/domain or nurtures the candidate's proficiency/skill is called an Elective Course.
 - 2.1 Discipline Specific Elective (DSE) Course:** Elective courses may be offered by the main discipline/subject of study is referred to as Discipline Specific Elective. The University/Institute may also offer discipline related Elective courses of interdisciplinary nature (to be offered by main discipline/subject of study).
 - 2.2 Dissertation/Project:** An elective course designed to acquire special/advanced knowledge, such as supplement study/support study to a project work, and a candidate studies such a course on his own with an advisory support by a teacher/faculty member is called dissertation/project.
 - 2.3 Generic Elective (GE) Course:** An elective course chosen generally from an unrelated discipline/subject, with an intention to seek exposure is called a Generic Elective.

P.S.: A core course offered in a discipline/subject may be treated as an elective by other discipline/subject and vice versa and such electives may also be referred to as Generic Elective.
- 3. Ability Enhancement Courses (AEC)/Competency Improvement Courses/Skill Development Courses/Foundation Course:** The Ability Enhancement (AE) Courses may be of two kinds: AE Compulsory Course (AECC) and AE Elective Course (AEEC). "AECC" courses are the courses based upon the content that leads to Knowledge enhancement. They ((i) Environmental Science, (ii) English/MIL Communication) are mandatory for all disciplines. AEEC courses are value-based and/or skill-based and are aimed at providing hands-on-training, competencies, skills, etc.
 - 3.1 AE Compulsory Course (AECC):** Environmental Science, English Communication/MIL Communication.
 - 3.2 AE Elective Course (AEEC):** These courses may be chosen from a pool of courses designed to provide value-based and/or skill-based instruction.

Project work/Dissertation is considered as a special course involving application of knowledge in solving / analyzing /exploring a real life situation / difficult problem. A Project/Dissertation work would be of 6 credits. A Project/Dissertation work may be given in lieu of a discipline specific elective paper.

Details of courses under B.A (Honors), B.Com (Honors) & B.Sc. (Honors)

Course	*Credits	
	Theory+ Practical	Theory + Tutorial
<u>I. Core Course</u>		
(14 Papers)	14X4= 56	14X5=70
Core Course Practical / Tutorial*		
(14 Papers)	14X2=28	14X1=14
<u>II. Elective Course</u>		
(8 Papers)		
A.1. Discipline Specific Elective	4X4=16	4X5=20
(4 Papers)		
A.2. Discipline Specific Elective		
Practical/ Tutorial*	4 X 2=8	4X1=4
(4 Papers)		
B.1. Generic Elective/		
Interdisciplinary	4X4=16	4X5=20
(4 Papers)		
B.2. Generic Elective		
Practical/ Tutorial*	4 X 2=8	4X1=4
(4 Papers)		
<ul style="list-style-type: none"> • Optional Dissertation or project work in place of one Discipline Specific Elective paper (6 credits) in 6th Semester 		
<u>III. Ability Enhancement Courses</u>		
1. Ability Enhancement Compulsory		
(2 Papers of 2 credit each)	2 X 2=4	2 X 2=4
Environmental Science		
English/MIL Communication		
2. Ability Enhancement Elective (Skill Based)		
(Minimum 2)	2 X 2=4	2 X 2=4
(2 Papers of 2 credit each)		
Total credit	140	140
Institute should evolve a system/policy about ECA/ General Interest/Hobby/Sports/NCC/NSS/related courses on its own.		
* wherever there is a practical there will be no tutorial and vice-versa		

PROPOSED SCHEME FOR CHOICE BASED CREDIT SYSTEM IN
B. Sc. Honours (Computer Science)

SE ME ST ER	CORE COURSE (14)	Ability Enhancement Compulsory Course(AECC) (2)	Skill Enhancemen t Course (SEC) (2)	Elective: Discipline SpecificDSE (4)	Elective:Ge neric (GE) (6)
I	Programming Fundamentals using C++				GE – 1
	Computer System Architecture				
II	Programming in JAVA				GE – 2
	Discrete Structures				
III	Data Structures		SEC – 1		GE – 3
	Operating Systems				
	Computer Networks				
IV	Design and Analysis of Algorithms		SEC – 2		GE – 4
	Software Engineering				
	Database Management Systems				
V	Internet Technologies			DSE – 1	
	Theory of Computation			DSE – 2	
VI	Artificial Intelligence			DSE – 3	
	Computer Graphics			DSE – 4	

SEMESTER	COURSE OPTED	COURSE NAME	Credits
I			
	Core course-I	Programming Fundamentals using C++	4
	Core Course-I Practical/Tutorial	Programming Fundamentals using C++ Lab	2
	Core course-II	Computer System Architecture	4
	Core Course-II Practical/Tutorial	Computer System Architecture Lab	2
	Generic Elective-1	GE-1	4/5
	Generic Elective-1 Practical/Tutorial		2/1
II			2
	Core course-III	Programming in JAVA	4
	Core Course-III Practical/Tutorial	Programming in JAVA Lab	2
	Core course-IV	Discrete Structures	5
	Core Course-IV Practical/Tutorial	Discrete Structures Tutorial	1
	Generic Elective-2	GE-2	4/5
	Generic Elective-2 Practical/Tutorial		2/1
III	Core course-V	Data Structures	4
	Core Course-V Practical/Tutorial	Data Structures Lab	2
	Core course-VI	Operating Systems	4
	Core Course-VI Practical/Tutorial	Operating Systems Lab	2
	Core course-VII	Computer Networks	4
	Core Course-VII Practical/Tutorial	Computer Networks Lab	2
	Skill Enhancement Course-1	SEC-1	2
	Generic Elective-3	GE-3	4/5
	Generic Elective-3 Practical/Tutorial		2/1
IV	Core course-VIII	Design and Analysis of Algorithms	4
	Course-VIII Practical/Tutorial	Design and Analysis of Algorithms Lab	2
	Core course-IX	Software Engineering	4
	Core Course-IX Practical/Tutorial	Software Engineering Lab	2
	Core course-X	Database Management Systems	4
	Core Course-X Practical/Tutorial	Database Management Systems Lab	2
	Skill Enhancement Course-2	SEC-2	2
	Generic Elective-4	GE-4	4/5
	Generic Elective-4 Practical		2/1
V	Core course-XI	Internet Technologies	4
	Core Course-XI Practical/Tutorial	Internet Technologies Lab	2
	Core course-XII	Theory of Computation	5
	Core Course-XII Practical/Tutorial	Theory of Computation Tutorial	1
	Discipline Specific Elective-1	DSE-1	4
	Discipline Specific Elective-1 Practical/Tutorial	DSE-1 Lab	2
	Discipline Specific Elective-2	DSE-2	4

	Discipline Specific Elective-1 Practical/Tutorial	DSE-2 Lab	2
VI	Core course-XIII	Artificial Intelligence	4
	Core Course-III Practical/Tutorial	Artificial Intelligence Lab	2
	Core course-XIV	Computer Graphics	4
	Core Course-XIV Practical/Tutorial	Computer Graphics Lab	2
	Discipline Specific Elective-3	DSE-3	4
	Discipline Specific Elective-3 Practical/Tutorial	DSE-3 Lab	2
	Discipline Specific Elective-4	DSE-4	4
Discipline Specific Elective-4 Practical/Tutorial	DSE-4 Lab	2	
Total Credits			140

Core Papers(C): (Credit: 06 each)(1 period / week for tutorials or 4 periods / week of practical)

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

Discipline Specific Elective Papers: (Credit: 06 each) – DSE- 1,DSE- 2 ,DSE- 3 , DSE- 4.

DSE – 1 (any one)

- a) System Programming
- b) Numerical Methods
- c) Operational Research for Computer Science

DSE – 2 (any one)

- a) Microprocessor
- b) Modeling and Simulation
- c) Advanced Algorithms

DSE – 3 (any one)

- a) Machine Learning
- b) Introduction to Data Sciences
- c) Combinatorial Optimization

DSE – 4 (any one)

- a) Digital Image Processing
- b) Data Mining
- c) Project Work / Dissertation

Note: Universities may include more options or delete some from this list

Other Discipline (Four papers of any one discipline) – GE 1 to GE 4

Any one discipline of importance

Skill Enhancement Courses (Credit: 02 each): SEC – 1, SEC – 2

SEC – 1

Android Programming

SEC – 2

PHP Programming

Note: Universities may include more options or delete some from this list

Important:

1. Each University/Institute should provide a brief write-up about each paper outlining the salient features, utility, learning objectives and prerequisites.
2. University/Institute can add/delete some experiments of similar nature in the Laboratory papers.
3. The size of the practical group for practical papers is recommended to be 10-15 students.
4. The size of tutorial group for papers without practical is recommended to be 8-10 students.
5. University/Institute can add to the list of reference books given at the end of each paper.

CORE COURSES (HONOURS IN COMPUTER SCIENCE)

Total Marks: 150

Theory: 75

Practical: 50

Internal Assessment: 25

5 Lectures , 4 Practicals(each in group of 15 to 20) ,

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

Theory: 60 Lectures

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

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

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

Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar etc), 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 Lectures)

Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

4. Functions and Arrays (10 Lectures)

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, Return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments.

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays

5. Derived Data Types (Structures and Unions) (3 Lectures)

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

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

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

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

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

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

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

9. Using Classes in C++

(7 Lectures)

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

10. Overview of Function Overloading and Operator Overloading

(5 Lectures)

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

11. Inheritance, Polymorphism and Exception Handling

(8 Lectures)

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

Reference Books

1. HerbtzSchildt, "C++: The Complete Reference", Fourth Edition, McGraw Hill.
2. BjarneStroustrup, "The C++ Programming Language", 4th Edition, Addison-Wesley , 2013.
3. BjarneStroustrup, "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.
9. Walter Savitch, "Problem Solving with C++", Pearson Education, 2007.
10. Stanley B. Lippman, JoseeLajoie, Barbara E. Moo, "C++ Primer", Published by Addison-Wesley, 5th Edition, 2012

COMPUTER SCIENCE LAB (C-I): Programming Fundamentals using C++ Lab

Practical: 60 Lectures

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

the prime numbers less than 100.

7. WAP to compute the factors of a given number.

8. Write a macro that swaps two numbers. WAP to use it.

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

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

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

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

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

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

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

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

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

15. Write a program to find sum of n elements entered by the user. To write this program, allocate memory dynamically using malloc() / calloc() functions or new operator.

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

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

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

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

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

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

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

- a) Sum b) Difference c) Product d) Transpose
22. 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:
- Calculate surface Area
 - Calculate Volume
 - Increment, Overload ++ operator (both prefix & postfix)
 - Decrement, Overload -- operator (both prefix & postfix)
 - Overload operator == (to check equality of two boxes), as a friend function
 - Overload Assignment operator
 - 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.

## **COMPUTER SCIENCE (C-II): Computer System Architecture**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### **1. Introduction**

**(8 lectures)**

Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexers, registers, counters and memory units.

### **2. Data Representation and Basic Computer Arithmetic**

**(10 lectures)**

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

### **3. Basic Computer Organization and Design**

**(13 lectures)**

Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Interconnection Structures, Bus Interconnection design of basic computer.

**4. Central Processing Unit (15 lectures)**

Register organization, arithmetic and logical micro-operations, stack organization, micro programmed control. Instruction formats, addressing modes, instruction codes, machine language, assembly language, input output programming, RISC, CISC architectures, pipelining and parallel architecture.

**5. Memory Organization (6 lectures)**

Cache memory, Associative memory, mapping.

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

Input / Output: External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access, I/O Channels.

**Recommended Books:**

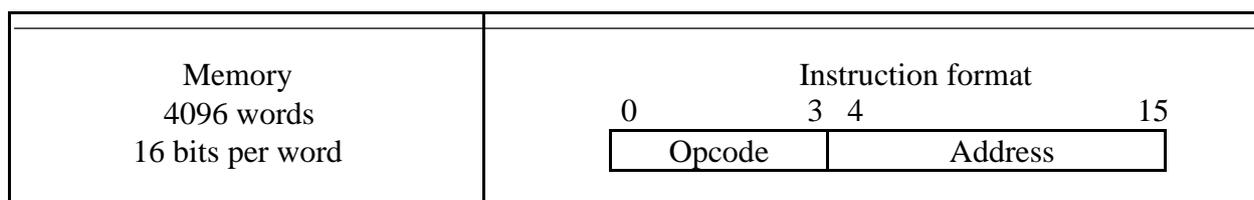
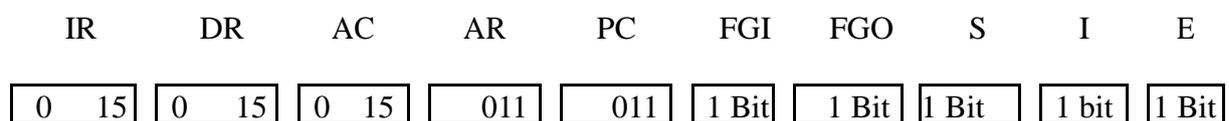
1. M. Mano, Computer System Architecture, Pearson Education 1992
2. A. J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology, 2004
3. W. Stallings, Computer Organization and Architecture Designing for Performance, 8<sup>th</sup> Edition, Prentice Hall of India, 2009
4. Digital Design, M.M. Mano, Pearson Education Asia

**COMPUTER SCIENCE LAB (C-II): Computer System Architecture Lab**

**Practical: 60 Lectures**

1. Create a machine based on the following architecture:

Register Set



Basic Computer Instructions

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

**Optional**

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

2. Create the micro operations and associate with instructions as given in the chapter (except interrupts). Design the register set, memory and the instruction set. Use this machine for the assignments of this section.
3. Create a Fetch routine of the instruction cycle.
4. Simulate the machine to determine the contents of AC, E, PC, AR and IR registers in hexadecimal after the execution of each of following register reference instructions:
  - a. CLA
  - b. CLE
  - c. CMA
  - d. CME
  - e. CIR
  - f. CIL
  - g. INC
  - h. SPA
  - i. SNA
  - j. SZA
  - k. SZE
  - 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
  - b. AND
  - c. LDA
  - d. STA
  - e. BUN
  - f. BSA
  - g. ISZ

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

**Instruction format**

|        |   |         |   |    |
|--------|---|---------|---|----|
| 0      | 2 | 3       | 4 | 15 |
| Opcode | I | Address |   |    |

- 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.

## COMPUTER SCIENCE (C-III): Programming in Java

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### 1. Introduction to Java

**(4 Lectures)**

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

### 2. Arrays, Strings and I/O

**(8 Lectures)**

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

Reading/Writing from console and files.

### **3. Object-Oriented Programming Overview**

**(4 Lectures)**

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

### **3. Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata (14 lectures)**

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

### **4. Exception Handling, Threading, Networking and Database Connectivity (15 Lectures)**

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

### **5. Applets and Event Handling**

**(15 Lectures)**

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

### **Reference Books**

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
2. James Gosling, Bill Joy, Guy L Steele Jr, Gilad Bracha, Alex Buckley "The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
3. Joshua Bloch, "Effective Java" 2nd Edition, Publisher: Addison-Wesley, 2008.
4. Cay S. Horstmann, Gary Corness, "Core Java 2 Volume 1 - Fundamentals)", 9th Edition, Prentice Hall.
5. Cay S. Horstmann, Gary Corness, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, Prentice Hall.
6. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
7. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.
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.

## Practical: 60 Lectures

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and StringBuffer classes like setCharAt(), setLength(), append(), insert(), concat() and equals().
9. Write a program to create a “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 Fibonacci series is given in a different file belonging to the same package.
17. Write a program that creates illustrates different levels of protection in classes/subclasses belonging to same package or different packages
18. Write a program “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 to create URL object, create a URL Connection using the openConnection() method and then use it examine the different components of the URL and content.
24. Write a program to implement a simple datagram client and server in which a message that is typed into the server window is sent to the client side where it is displayed.
25. Write a program that creates a Banner and then creates a thread to scrolls the message in the banner from left to right across the applet’s window.
26. Write a program to get the URL/location of code (i.e. java code) and document (i.e. html file).
27. Write a program to demonstrate different mouse handling events like mouseClicked(), mouseEntered(), mouseExited(), mousePressed, mouseReleased() and mouseDragged().
28. Write a program to demonstrate different keyboard handling events.
29. Write a program to generate a window without an applet window using main() function.

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

---

## **COMPUTER SCIENCE (C-IV): Discrete Structures**

**Total Marks: 100**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

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

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

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

Asymptotic Notations, Summation formulas and properties, BoundingSummations, approximation by Integrals

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

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

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

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

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

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

### **Recommended Books:**

1. C.L. Liu & Mahopatra, Elements of Discrete mathematics, 2<sup>nd</sup> Sub Edition 1985, Tata McGraw Hill
2. Rosen, Discrete Mathematics and Its Applications, Sixth Edition 2006
3. T.H. Cormen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, Prentice Hall on India (3<sup>rd</sup> edition 2009)
4. M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms 1988 Johnwiley Publication

5. J. L. Hein, Discrete Structures, Logic, and Computability, Jones and Bartlett Publishers, 3rd Edition, 2009
6. D.J. Hunter, Essentials of Discrete Mathematics, Jones and Bartlett Publishers, 2008

## **COMPUTER SCIENCE LAB (C-IV): Discrete Structures Practicals**

### **Practicals based on the theory**

---

## **COMPUTER SCIENCE (C-V): Data Structures**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

1. **Arrays** **(5 Lectures)**  
Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation)
2. **Stacks** **(5 Lectures)**  
Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack
3. **Linked Lists** **(10 Lectures)**  
Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists
4. **Queues** **(5 Lectures)**  
Array and Linked representation of Queue, De-queue, Priority Queues
5. **Recursion** **(5 lectures)**  
Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)
6. **Trees** **(20 Lectures)**  
Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion , Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees).
7. **Searching and Sorting** **(5 Lectures)**  
Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques
8. **Hashing** **(5 Lectures)**  
Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table

Reordering, Resolving collision 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.
  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; 2 edition, 2009
  10. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley
  11. Herbert Schildt, "Java The Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.
  12. D. S. Malik, P.S. Nair, "Data Structures Using Java", Course Technology, 2003.
- 

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

##### **Practical: 60 Lectures**

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.

11. WAP to calculate factorial and to compute the factors of a given no. (i) using recursion, (ii) using iteration
  12. (ii) WAP to display fibonacci series (i) using recursion, (ii) using iteration
  13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
  14. WAP to create a Binary Search Tree and include following operations in tree:
    - (a) Insertion (Recursive and Iterative Implementation)
    - (b) Deletion by copying
    - (c) Deletion by Merging
    - (d) Search a no. in BST
    - (e) Display its preorder, postorder and inorder traversals Recursively
    - (f) Display its preorder, postorder and inorder traversals Iteratively
    - (g) Display its level-by-level traversals
    - (h) Count the non-leaf nodes and leaf nodes
    - (i) Display height of tree
    - (j) 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.
- 

## **COMPUTER SCIENCE (C-VI): Operating Systems**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### **1.Introduction**

(10 Lectures)

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

### **2.Operating System Organization**

(6 Lectures)

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

### **3.Process Management**

( 20Lectures)

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

#### **4.Memory Management** (10 Lectures)

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

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

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

#### **6.Protection and Security** (4 Lectures)

Policy mechanism, Authentication, Internal access Authorization.

#### **Recommended Books:**

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

---

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

#### **Practical: 60 Lectures**

##### **C/ C++ programs**

1. WRITE A PROGRAM (using *fork()* and/or *exec()* commands) where parent and child execute:
  - a) same program, same code.
  - b) same program, different code.
  - c) before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behaviour of Linux kernel including information on

configured memory, amount of free and used memory. (memory information)

4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.
  5. WRITE A PROGRAM to copy files using system calls.
  6. Write program to implement FCFS scheduling algorithm.
  7. Write program to implement Round Robin scheduling algorithm.
  8. Write program to implement SJF scheduling algorithm.
  9. Write program to implement non-preemptive priority based scheduling algorithm.
  10. Write program to implement preemptive priority based scheduling algorithm.
  11. Write program to implement SRJF scheduling algorithm.
  12. Write program to calculate sum of n numbers using *thread* library.
  13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.
- 

## **COMPUTER SCIENCE (C-VII): Computer Networks**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### **1. Introduction to Computer Networks**

(8 Lectures)

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

### **2. Data Communication Fundamentals and Techniques**

(10 Lectures)

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

### **3. Networks Switching Techniques and Access mechanisms**

(10 Lectures)

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

### **4. Data Link Layer Functions and Protocol**

(10 Lectures)

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

**5. Multiple Access Protocol and Networks** (5 Lectures)

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

**6. Networks Layer Functions and Protocols** (6 Lectures)

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

**7. Transport Layer Functions and Protocols** (6 Lectures)

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

**8. Overview of Application layer protocol** (5 Lectures)

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

**Reference Books**

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

**COMPUTER SCIENCE LAB (C-VII): Computer Networks Lab**

**Practical: 60 Lectures**

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

---

**COMPUTER SCIENCE (C-VIII): Design and Analysis of Algorithms**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

**1.Introduction** ( 5 Lectures)  
Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm.

**2.Algorithm Design Techniques** ( 8 Lectures)  
Iterative techniques, Divide and Conquer, DynamicProgramming, Greedy Algorithms.

**3.Sorting and Searching Techniques** (20 Lectures)  
Elementary sorting techniques–Bubble Sort, InsertionSort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis;

**4.Lower Bounding Techniques** ( 5 Lectures)  
Decision Trees

**5.Balanced Trees** ( 7 Lectures)  
Red-Black Trees

**6.Advanced Analysis Technique** ( 5 Lectures)  
Amortized analysis

**7.Graphs** (5 Lectures)  
Graph Algorithms–Breadth First Search, Depth First Search and its Applications,Minimum Spanning Trees.

**8.String Processing** (5Lectures)  
String Matching, KMP Technique

**Recommended Books:**

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

**COMPUTER SCIENCE LAB (C-VIII): Design and Analysis of Algorithms Lab**

**Practical: 60 Lectures**

1. i. Implement Insertion Sort (The program should report the number of comparisons)  
ii. Implement Merge Sort(The program should report the number of comparisons)
2. Implement Heap Sort(The program should report the number of comparisons)

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

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  $n \log n$ .

## **COMPUTER SCIENCE (C-IX): Software Engineering**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### **1.Introduction**

(8 Lectures)

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

### **2.Requirement Analysis**

(10 Lectures)

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

### **3.Software Project Management**

(8 Lectures)

Estimation in Project Planning Process, Project Scheduling.

### **4.Risk Management**

(8 Lectures)

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

### **5.Quality Management**

(8 Lectures)

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

### **6.Design Engineering**

(10 Lectures)

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 Lectures)

Software Testing Fundamentals, Strategic Approach to Software Testing, Test Strategies for Conventional Software, Validation Testing, System Testing, Black-Box Testing, White-Box Testing and their type, Basis Path Testing.

### Recommended Books:

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

## COMPUTER SCIENCE LAB (C-IX): Software Engineering Lab

### Practical: 60 Lectures

| S. No | Practical Title                                                                                                                                    |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.    | <ul style="list-style-type: none"><li>• Problem Statement,</li><li>• Process Model</li></ul>                                                       |
| 2.    | Requirement Analysis: <ul style="list-style-type: none"><li>• Creating a Data Flow</li><li>• Data Dictionary, Use Cases</li></ul>                  |
| 3.    | Project Management: <ul style="list-style-type: none"><li>• Computing FP</li><li>• Effort</li><li>• Schedule, Risk Table, Timeline chart</li></ul> |
| 4.    | Design Engineering: <ul style="list-style-type: none"><li>• Architectural Design</li><li>• Data Design, Component Level Design</li></ul>           |
| 5.    | Testing: <ul style="list-style-type: none"><li>• Basis Path Testing</li></ul>                                                                      |

### 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

## **COMPUTER SCIENCE (C-X): Database Management Systems**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

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

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

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

Entity types, relationships, constraints.

### **3.Relation data model** (20 Lectures)

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

### **4.Database design** (15 Lectures)

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

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

ACID properties, concurrency control

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

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<sup>+</sup> trees.

**Books Recommended:**

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

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

### Practical: 60 Lectures

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

#### EMPLOYEE Schema

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

#### DEPARTMENT Schema

| Field    | Type        | NULL | KEY | DEFAULT   |
|----------|-------------|------|-----|-----------|
| Dno      | Integer     | No   | PRI | NULL      |
| Dname    | Varchar(50) | Yes  |     | NULL      |
| Location | Varchar(50) | Yes  |     | New Delhi |

#### Query List

1. Query to display Employee Name, Job, Hire Date, Employee Number; for each employee with the Employee Number appearing first.
2. Query to display unique Jobs from the Employee Table.
3. Query to display the Employee Name concatenated by a Job separated by a comma.

4. Query to display all the data from the Employee Table. Separate each Column by a comma and name the said column as THE\_OUTPUT.
5. Query to display the Employee Name and Salary of all the employees earning more than \$2850.
6. Query to display Employee Name and Department Number for the Employee No= 7900.
7. Query to display Employee Name and Salary for all employees whose salary is not in the range of \$1500 and \$2850.
8. Query to display Employee Name and Department No. of all the employees in Dept 10 and Dept 30 in the alphabetical order by name.
9. Query to display Name and Hire Date of every Employee who was hired in 1981.
10. Query to display Name and Job of all employees who don't have a current Manager.
11. Query to display the Name, Salary and Commission for all the employees who earn commission.
12. Sort the data in descending order of Salary and Commission.
13. Query to display Name of all the employees where the third letter of their name is 'A'.
14. Query to display Name of all employees either have two 'R's or have two 'A's in their name and are either in Dept No = 30 or their Manger's Employee No = 7788.
15. Query to display Name, Salary and Commission for all employees whose Commission Amount is 14 greater than their Salary increased by 5%.
16. Query to display the Current Date.
17. Query to display Name, Hire Date and Salary Review Date which is the 1st Monday after six months of employment.
18. Query to display Name and calculate the number of months between today and the date each employee was hired.
19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 \* Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with 'J', 'A' and 'M'.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.

23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an 'A' in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees' Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (\*) signifies \$100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a 'T'.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department.

---

## **COMPUTER SCIENCE (C-XI): Internet Technologies**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

**Java****(5 lectures)**

Use of Objects, Array and ArrayList class

**JavaScript****(15 lectures)**

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

**JDBC****(10 lectures)**

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

**JSP****(20 lectures)**

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 lectures)**

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

**Recommended Books:**

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

**COMPUTER SCIENCE LAB (C-XI): Internet Technologies Lab****Practical: 60 Lectures**

Create event driven program for following:

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

---

## **COMPUTER SCIENCE (C-XII): Theory of Computation**

**Total Marks: 100**

**Theory: 75**

**Internal Assessment: 25**

**5 Lectures , 1 Tutorial.**

**Theory: 60 Lectures**

### **1. Languages** (8 Lectures)

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

### **2. Finite Automata and Regular Languages** (20 Lectures)

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.

### **3. Context free languages** (17 Lectures)

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

### **4. Turing Macines and Models of Computations** (15 Lectures)

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

#### **Recommended Books:**

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

## **COMPUTER SCIENCE LAB (C-XII): Theory of Computation**

**Tutorial**

---

## **COMPUTER SCIENCE (C-XIII): Artificial Intelligence**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### **1.Introduction**

(06 Lectures)

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

### **2.Problem Solving and Searching Techniques**

(20 Lectures)

Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A\* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.

### **3.Knowledge Representation**

(20 Lectures)

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

Programming in Logic (PROLOG)

### **4.Dealing with Uncertainty and Inconsistencies**

(08 Lectures)

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

### **5.Understanding Natural Languages**

(06 Lectures)

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

### **BOOKS RECOMMENDED:**

1. DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI, 2007.
2. Russell &Norvig, Artificial Intelligence-A Modern Approach, LPE, Pearson Prentice Hall, 2<sup>nd</sup> edition, 2005.
3. Rich & Knight, Artificial Intelligence – Tata McGraw Hill, 2<sup>nd</sup> edition, 1991.
4. W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House, 3<sup>rd</sup> edition, 2001.

5. Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education, 3<sup>rd</sup> edition, 2000.

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

### **Practical: 60 Lectures**

1. Write a prolog program to calculate the sum of two numbers.
2. Write a prolog program to find the maximum of two numbers.
3. Write a prolog program to calculate the factorial of a given number.
4. Write a prolog program to calculate the nth Fibonacci number.
5. Write a prolog program, insert\_nth(item, n, into\_list, result) that asserts that result is the list into\_list with item inserted as the n'th element into every list at all levels.
6. Write a Prolog program to remove the Nth item from a list.
7. Write a Prolog program, remove\_nth(Before, After) that asserts the After list is the Before list with the removal of every n'th item from every list at all levels.
8. Write a Prolog program to implement append for two lists.
9. Write a Prolog program to implement palindrome(List).
10. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
11. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
12. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement reverse(List,ReversedList) that reverses lists.
15. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.
16. Write a Prolog program to implement GCD of two numbers.
17. Write a prolog program that implements Semantic Networks/Frame Structures.

---

## **COMPUTER SCIENCE LAB (C-XIV): Computer Graphics**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### **1.Introduction** (5 Lectures)

Basic elements of Computer graphics, Applications of Computer Graphics.

### **2.Graphics Hardware** (8 Lectures)

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

### **3.Fundamental Techniques in Graphics** (22 Lectures)

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

### **4.Geometric Modeling** (10 Lectures)

Representing curves & Surfaces.

### **5.Visible Surface determination** (8 Lectures)

Hidden surface elimination.

### **6.Surface rendering** (7 Lectures)

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

#### **Books Recommended:**

1. J.D.Foley, A.Van Dan, Feiner, Hughes Computer Graphics Principles & Practice 2<sup>nd</sup> 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 2<sup>nd</sup> edition 1989.

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

### **Practical: 60 Lectures**

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

---

**Discipline Specific Elective Papers COMPUTER SCIENCE: (Credit: 06 each) – DSE-1 ,DSE-2,DSE-3,DSE- 4.**

**DSE – 1 (any one)**

---

### **1 (a) Systems Programming**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 lectures**

**Assemblers & Loaders, Linkers:**

12L

One pass and two pass assembler, design of an assembler, Absolute loader, relocation and linking concepts, relocating loader and Dynamic Linking.

**Introduction:**

Overview of compilation, Phases of a compiler 2L

**Lexical Analysis:**

Role of a Lexical analyzer, Specification and recognition of tokens, Symbol table, lex 10L

**Parsing:**

Bottom up parsing- LR parser, yacc. 15L

**Intermediate representations**

Three address code generation, syntax directed translation, translation of types, control statements 10L

**Storage organization:**

Activation records, stack allocation 6L

**Code Generation:**

Object code generation 5L

**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*, 2<sup>nd</sup> edition, PrenticeHall, 2006.
3. D. M. Dhamdhare, *Systems Programming*, Tata McGraw Hill, 2011.
4. Leland Beck, D. Manjula, *System Software: An Introduction to System Programming*, 3<sup>rd</sup> edition, Pearson Education, 2008.
5. Grune D, Van Reeuwijk . K, Bal H. E, Jacobs C J H, Langendoen K, *Modern CompilerDesign*, 2<sup>nd</sup> edition, Springer, 2012

**SYSTEMS PROGRAMMING LAB****Practical: 60 lectures**

[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.

---

## 1 (b) Numerical Methods

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

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 6L

Bisection method, Secant method, Regula–Falsi method  
Newton–Raphson method, Newton’s method for solving nonlinear systems 10L

Gauss elimination method (with row pivoting) and Gauss–Jordan method, Gauss Thomas method for tridiagonal systems  
Iterative methods: Jacobi and Gauss-Seidel iterative methods  
10L

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),  
10L

Numerical differentiation: First derivatives and second order derivatives, Richardson extrapolation  
10L

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<sup>th</sup> order Runge-Kutta method, Finite difference method for linear ODE  
14L

### 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)

### Numerical Methods Lab

## Practical: 60 lectures

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

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

---

## 1(c) Operational Research for Computer Science

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

### Introductory Linear Algebra

**6 hrs**

System of linear equations, Matrices, Rank and Determinant of a matrix, Linearly dependent and independent vectors, Basis of a matrix.

### Linear programming - I

**10 hrs**

Optimization Problems, Introduction to LP Formulation, Convex sets, Extreme points, Geometry of Linear Programs, Basic feasible solutions (BFS), Neighborhoods, Local and global optima, Profitable Column, Pivoting, Simplex Algorithm with initial BFS, Graphical method

### Linear programming - II

**10 hrs**

Degeneracy and Bland's Anticycling rule (Definition), Simplex Algorithm without initial BFS, Artificial variable techniques – two phase method, M-Charnes method, special cases in LPP.

### Duality

**10 hrs**

Definition of the dual problem, primal-dual relationships, economic interpretation of duality, complementary slackness conditions.

### Transportation Models

**6 hrs**

Transportation Algorithm, Assignment model, Hungarian Method

**Introduction to Queuing Models**

**9 hrs**

Elements of Queuing Model, Exponential distribution, Poisson Distributions, Poisson Queuing Models, Single Server model, Multiple Server model

**Introduction to Markov Chains**

**9 hrs**

Introduction to Markov chains, transition probabilities, classification of states, Steady state probabilities, Absorbing states

**Reference Books**

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

**Practical: 60 Lectures**

**Practical based on topics covered in the paper**

**DSE – 2 (any one)**

---

**2(a) Microprocessor**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

**Microprocessor architecture:** Internal architecture, system bus architecture, memory and I/O interfaces. 15L

**Microprocessor programming:** Register Organization, instruction formats, assembly language programming. 15L

**Interfacing:** Memory address decoding, cache memory and cache controllers, I/O interface, keyboard, display, timer, interrupt controller, DMA controller, video controllers, communication interfaces. 30L

**Recommended Books:**

1. Barry B. Brey : The Intel Microprocessors : Architecture, Programming and Interfacing.

- Pearson Education, Sixth Edition.2009
2. 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

#### **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 ascii conversion
  6. Write a program for ascii to binary conversion
- 
- 

## **2 (b) Modelling and Simulation**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 lectures**

### **System models and System studies**

**[5 Lectures]**

Concept of a System, Deterministic and Stochastic Activities, Continuous and Discrete Systems, System Modeling, Types of Models, Principles used in Modeling, Corporate Model, System Design.

### **What is Simulation?**

**[7 Lectures]**

Technique of Simulation, The Monte-Carlo Method, Comparison of Simulation and Analytical Methods, Experimental nature of Simulation, Types of System Simulation, Numerical Computation Techniques for Continuous Models, Numerical Computation Techniques for Discrete Models, Distributed Lag Models, Cobweb Models.

### **Continuous System Simulation**

**[7 Lectures]**

Continuous System Models, Differential Equations & Applications, Feedback Systems, Simulation of an Autopilot, Interactive systems, Real Time Systems.

### **Concepts in Discrete Event Simulation**

**[7 Lectures]**

The Event Scheduling / Time Advance Algorithm, World Views, Manual Simulation using Event Scheduling, List Processing: Lists-Basic properties and operations, Use of arrays for List Processing, Using Dynamic Allocation and Linked Lists.

### **Queuing Models**

**[9 Lectures]**

Characteristics of Queuing Systems, Arrival and Service Patterns, Queue Discipline, Long Run Measures of Performance of Queuing Systems, Time-Average Number in System, Server

Utilization, Costs in Queuing Problems, Steady State behavior of Infinite Population Markovian Models, Multiserver Queue:  $M/M/C/\infty/\infty$ .

### **Simulation software**

**[7 Lectures]**

Comparison of Simulation Packages with Programming Languages, Classification of Simulation Software: General Purpose vs Application Oriented Simulation Packages, Desirable Software Features: General Capabilities, H/w and S/w Requirements, Animation and Dynamic Graphics, Statistical Capabilities, General Purpose Simulation Packages, Object Oriented Simulation, Examples of Application-oriented Simulation packages, Simulation in GPSS.

### **Random Number, Non-Uniform random variate Generation and Monte-Carlo Method**

**[9 Lectures]**

Linear Congruential Generators, Testing Random Number Generators: Empirical and Theoretical tests, Non-Uniform Random Variate Generator: Inverse Transform, Composition. Generating Continuous Random Variates: Uniform, Exponential, Gamma, and Normal. Generating Discrete Random Variates: Bernoulli, Binomial, Poisson. Monte-Carlo Method: Evaluation of Integral-Hit or Miss Method.

### **Analysis of Simulation Data**

**[5 Lectures]**

Identifying the Distribution with Data, Types of Simulations with respect to Output Analysis, Stochastic nature of Output Data, Measures of Performance and their Estimation: Point Estimation, Confidence-Interval Estimation, Output Analysis for Terminating Simulations: Statistical Background, Confidence-Intervals with Specified Precision, Output Analysis for Steady State Simulations, Variance-Reduction Technique-Antithetic variates.

### **Verification and Validation of Simulation Models**

**[4 Lectures]**

Model Building- Verification and Validation, Verification of Simulation Models, Calibration and Validation of Models: Validation of Model Assumptions, Validating Input-output Transformations, Input-Output Validation

### **References:**

1. Raj Jain, Art of Computer Systems Performance Analysis, John Wiley and Sons, Inc, 1991
2. Sheldon M. Ross, Simulation, 4<sup>th</sup> Ed., Elsevier 2008
3. Averill M. Law and W. David Kelton, Simulation Modeling and Analysis, 3<sup>rd</sup> Ed., Tata McGraw-Hill, 2003
4. Geoferey Gordon, System Simulation, 2<sup>nd</sup> Ed., PHI, 1987
5. Jerry Banks and John S. Carson, Barry L Nelson, Discrete-Event System Simulation, 5<sup>th</sup> Ed., Prentice Hall, 2010
6. Narsingh Deo, System Simulation with Digital Computers, Prentice Hall of India, 1979

### **Modelling and Simulation Lab**

#### **Practicals: 60 lectures**

Practical exercises based on theory

## 2(c) Advanced Data Structures and Algorithms

**Total Marks: 100**

**Theory: 75**

**Internal Assessment: 25**

**5 Lectures , 1 Tutorial.**

Theory : 60

**Advanced Data Structures** **10 hrs**

Fibonacci heaps, Priority Queues. Dynamic Data Structures.

**Divide and Conquer** **8 hrs**

Closest pair of points, Integer Multiplication, Convolutions and Fast Fourier Transforms

**Greedy** **6 hrs**

Interval Scheduling, Proving optimality using Stays Ahead and Exchange Arguments.

**Dynamic Programming** **8 hrs**

Principles of Dynamic Programming, Weighted Interval Scheduling, Segmented Least Squares, Subset Sums and Knapsacks

**Network Flows** **10 hrs**

Max-flow problem, Ford Fulkerson Algorithm, Maximum flows and Minimum Cuts in a network, Bipartite Matching.

**NP Completeness** **18 hrs**

Polynomial time reductions, Reduction via gadgets (satisfiability problem), Efficient Certification and Definition of NP, NP Complete problems, Sequencing problems, Partitioning problems, co-NP and asymmetry of NP.

### **Books Recommended:**

1. Jon Kleinberg and Eva Tardos, Algorithm Design, Pearson Education, 2006
2. Cormen, Leiserson, Rivest and Stein, Introduction to Algorithms, Prentice Hall of India, Third Edition
3. Vijay V. Vazirani, Approximation algorithms, Springer

Tutorials

**DSE – 3 (any one)**

**3 (a) Machine Learning**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

**Introduction (8 lectures)**

Concept of Machine Learning, Applications of Machine Learning, Key elements of Machine Learning, Supervised vs. Unsupervised Learning, Statistical Learning: Bayesian Method, The Naive Bayes Classifier

**Softwares for Machine Learning and Linear Algebra Overview (12 lectures)**

Plotting of Data, Vectorization, Matrices and Vectors: Addition, Multiplication, Transpose and Inverse using available tool such as MATLAB.

**Linear Regression (10 lectures)**

Prediction using Linear Regression, Gradient Descent, Linear Regression with one variable, Linear Regression with multiple variables, Polynomial Regression, Feature Scaling/Selection.

**Logistic Regression (10 lectures)**

Classification using Logistic Regression, Logistic Regression vs. Linear Regression, Logistic Regression with one variable and with multiple variables.

**Regularization (10 lectures)**

Regularization and its utility: The problem of Overfitting, Application of Regularization in Linear and Logistic Regression, Regularization and Bias/Variance.

**Neural Networks (10 lectures)**

Introduction, Model Representation, Gradient Descent vs. Perceptron Training, Stochastic Gradient Descent, Multilayer Perceptrons, Multiclass Representation, Backpropagation Algorithm.

### **Recommended Books**

1. Ethem Alpaydin, "Introduction to Machine Learning" 2nd Edition, The MIT Press, 2009.
2. Tom M. Mitchell, "Machine Learning", First Edition by Tata McGraw-Hill Education, 2013.
3. Christopher M. Bishop, "Pattern Recognition and Machine Learning" by Springer, 2007.
4. Mevin P. Murphy, "Machine Learning: A Probabilistic Perspective" by The MIT Press, 2012.

## Machine Learning Lab

### Practical: 60 Lectures

For practical Labs for Machine Learning, students may use softwares like MABLAB/Octave or Python. For later exercises, students can create/use their own datasets or utilize datasets from online repositories like UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/>).

1. Perform elementary mathematical operations in Octave/MATLAB like addition, multiplication, division and exponentiation.
2. Perform elementary logical operations in Octave/MATLAB (like OR, AND, Checking for Equality, NOT, XOR).
3. Create, initialize and display simple variables and simple strings and use simple formatting for variable.
4. Create/Define single dimension / multi-dimension arrays, and arrays with specific values like array of all ones, all zeros, array with random values within a range, or a diagonal matrix.
5. Use command to compute the size of a matrix, size/length of a particular row/column, load data from a text file, store matrix data to a text file, finding out variables and their features in the current scope.
6. Perform basic operations on matrices (like addition, subtraction, multiplication) and display specific rows or columns of the matrix.
7. Perform other matrix operations like converting matrix data to absolute values, taking the negative of matrix values, adding/removing rows/columns from a matrix, finding the maximum or minimum values in a matrix or in a row/column, and finding the sum of some/all elements in a matrix.
8. Create various type of plots/charts like histograms, plot based on sine/cosine function based on data from a matrix. Further label different axes in a plot and data in a plot.
9. Generate different subplots from a given plot and color plot data.
10. Use conditional statements and different type of loops based on simple example/s.
11. Perform vectorized implementation of simple matrix operation like finding the transpose of a matrix, adding, subtracting or multiplying two matrices.
12. Implement Linear Regression problem. For example, based on a dataset comprising of existing set of prices and area/size of the houses, predict the estimated price of a given house.
13. Based on multiple features/variables perform Linear Regression. For example, based on a number of additional features like number of bedrooms, servant room, number of balconies, number of houses of years a house has been built – predict the price of a house.
14. Implement a classification/ logistic regression problem. For example based on different features of students data, classify, whether a student is suitable for a particular activity. Based on the available dataset, a student can also implement another classification problem like checking whether an email is spam or not.
15. Use some function for regularization of dataset based on problem 14.
16. Use some function for neural networks, like Stochastic Gradient Descent or backpropagation - algorithm to predict the value of a variable based on the dataset of problem 14.

---

### 3 (b) Introduction to Data Sciences

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

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

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

**Getting and Cleaning Data:** Obtaining data from the web, from APIs, from databases and from colleagues in various formats. basics of data cleaning and making data "tidy". 12L

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

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

### **Reference Books**

1. Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontline" by Schroff/O'Reilly, 2013.
2. Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking" by O'Reilly, 2013.
3. John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.
4. Ian Ayres, "Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart" Ist Edition by Bantam, 2007.
5. Eric Seigel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1<sup>st</sup>Edition, by Wiley, 2013.
6. Matthew A. Russel, "Mining the Social Web: Data mining Facebook, Twitter, LinkedIn, Goole+, GitHub, and More", Second Edition, by O'Reilly Media, 2013.

### **Introduction to Data Sciences Lab**

**Practical: 60 Lectures**

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 linear search.
8. Implement binary search.
9. Implement matrices addition , subtraction and Multiplication
10. Fifteen students were enrolled in a course. Their ages were:  
20 20 20 20 20 21 21 21 21 22 22 22 22 23 23 23
  - i. Find the median age of all students under 22 years
  - ii. Find the median age of all students
  - iii. Find the mean age of all students
  - iv. Find the modal age for all students
  - v. Two more students enter the class. The age of both students is 23. What is now mean, mode and median ?
11. Following table gives a frequency distribution of systolic blood pressure. Compute all the measures of dispersion.

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

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

### 3(c) Combinatorial Optimization

**Total Marks: 100**

**Theory: 75**

**Internal Assessment: 25**

**5 Lectures , 1 Tutorial**

**Theory (60 Lectures)**

#### **Algorithmic Perspective to Simplex Method**

**15 hrs**

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

#### **Integer Programming**

**10 hrs**

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

#### **Primal-Dual Algorithms**

**15 hrs**

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.

### **Network and Flow Algorithms**

**20 hrs**

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**

---

## **DSE – 4 (any one)**

---

### **4 (a) Digital Image Processing**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 Lectures**

#### **1. Introduction**

**(6 Lectures)**

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

#### **2. Spatial Domain Filtering**

**(7 Lectures)**

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

#### **3. Filtering in the Frequency domain**

**(8 Lectures)**

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

#### **4. Image Restoration**

**(8 Lectures)**

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

#### **5. Image Compression**

**(10 Lectures)**

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

#### **6. Wavelet based Image Compression**

**(5 Lectures)**

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

#### **7. Morphological Image Processing**

**(7 Lectures)**

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

#### **8. Image Segmentation**

**(9 Lectures)**

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

#### **Reference Books**

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

#### **Digital Image Processing Lab**

##### **Practical: 60 Lectures**

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

- i. Flow control and LOOP in SCILAB
  - j. Write given 2-D data in image file
2. To write and execute image processing programs using point processing method
    - a. Obtain Negative image
    - b. Obtain Flip image
    - c. Thresholding
    - d. Contrast stretching
  3. To write and execute programs for image arithmetic operations
    - a. Addition of two images
    - b. Subtract one image from other image
    - c. Calculate mean value of image
    - d. Different Brightness by changing mean value
  4. To write and execute programs for image logical operations
    - a. AND operation between two images
    - b. OR operation between two images
    - c. Calculate intersection of two images
    - d. Water Marking using EX-OR operation
    - e. NOT operation (Negative image)
  5. To write a program for histogram calculation and equalization using
    - a. Standard MATLAB function
    - b. Program without using standard MATLAB functions
    - c. C Program
  6. To write and execute program for geometric transformation of image
    - a. Translation
    - b. Scaling
    - c. Rotation
    - d. Shrinking
    - e. Zooming
  7. To understand various image noise models and to write programs for
    - a. image restoration
    - b. Remove Salt and Pepper Noise
    - c. Minimize Gaussian noise
    - d. Median filter and Weiner filter
  8. Write and execute programs to remove noise using spatial filters
    - a. Understand 1-D and 2-D convolution process
    - b. Use 3x3 Mask for low pass filter and high pass filter
  9. Write and execute programs for image frequency domain filtering
    - a. Apply FFT on given image
    - b. Perform low pass and high pass filtering in frequency domain
    - c. Apply IFFT to reconstruct image
  10. Write a program in C and MATLAB/SCILAB for edge detection using different edge detection mask
  11. Write and execute program for image morphological operations erosion and dilation.
  12. To write and execute program for wavelet transform on given image and perform inverse

wavelet transform to reconstruct image.

---

#### **4 (b) Data Mining**

**Total Marks: 150**

**Theory: 75**

**Practical: 50**

**Internal Assessment: 25**

**5 Lectures , 4 Practicals( each in group of 15 to 20)**

**Theory: 60 lectures**

Introduction : Basic Data Mining Tasks, Data Mining Issues, Data Mining Metrics, Data Mining from a Database Perspective. 6L

Data Mining Techniques : A Statistical Perspective on Data Mining, Similarity Measures, Decision Trees, Neural Networks, Genetic Algorithms. 10L

Classification : Statistical-Based Algorithms, Distance-Based Algorithms, Decision Tree-Based Algorithms, Neural Network-Based Algorithms, Rule-Based Algorithms, Combining Techniques. 15L

Clustering : Similarity and Distance Measures, Hierarchical Algorithms, Partitional Algorithms, Clustering Large Databases, Clustering with Categorical Attributes. 12L

Association Rules : Basic Algorithms, Parallel and Distributed Algorithms, Incremental Rules, Advanced Association Rule Techniques, Measuring the Quality of Rules. 12L

Advanced Techniques : Web Mining, Spatial Mining, Temporal Mining. 5L

#### **Data Mining Lab**

**Practical: 60 lectures**

Practical exercises based on concepts listed in theory.

---

#### **4 (c) Project work / Dissertation**

**Total Marks: 150**

**Project:100**

**Internal Assessment: 50**

**Total Marks: 150**

**5 Lectures**

This option is to be offered only in 6<sup>th</sup> Semester.

The students will be allowed to work on any project based on the concepts studied in core / elective

or skill based elective courses.

The group size should be maximum of three (03) students.

Each group will be assigned a teacher as a supervisor who will handle both their theory as well lab classes.

A maximum of Four (04) projects would be assigned to one teacher.

---

## **Skill Enhancement Courses (Credit: 02 each) – SEC1 , SEC2 SEC – 1**

**Total Marks: 75**

**Practical: 50**

**Internal Assessment: 25**

**2 Lectures , 2 Practicals ( each in group of 15 to 20)**

### **Android Programming**

**Introduction:** History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture. (2L)

**Overview of object oriented programming using Java:** OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine. (3L)

**Development Tools:** Installing and using Eclipse with ADT plug-in, Installing Virtual machine for Android sandwich/Jelly bean (Emulator), configuring the installed tools, creating a android project – Hello Word, run on emulator, Deploy it on USB-connected Android device. (4L)

**User Interface Architecture:** Application context, intents, Activity life cycle, multiple screen sizes. (2L)

**User Interface Design:** Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners(Combo boxes),Images, Menu, Dialog. (2L)

**Database:** Understanding of SQLite database, connecting with the database. (2L)

### **Book Recommended:**

1. Android application development for java programmers. By James C. Sheusi. Publisher: Cengage Learning, 2013.

### **ONLINE READING / SUPPORTING MATERIAL:**

1. <http://www.developer.android.com>
2. <http://developer.android.com/about/versions/index.html>
3. <http://developer.android.com/training/basics/firstapp/index.html>
4. <http://docs.oracle.com/javase/tutorial/index.htm> (Available in the form of free downloadable ebooks also).
5. <http://developer.android.com/guide/components/activities.html>
6. <http://developer.android.com/guide/components/fundamentals.html>
7. <http://developer.android.com/guide/components/intents-filters.html>.
8. <http://developer.android.com/training/multiscreen/screensizes.html>
9. <http://developer.android.com/guide/topics/ui/controls.html>
10. <http://developer.android.com/guide/topics/ui/declaring-layout.html>

11. <http://developer.android.com/training/basics/data-storage/databases.html>

### **Software Lab Based on Android Programming:**

1. Create “Hello World” application. That will display “Hello World” in the middle of the screen in the emulator. Also display “Hello World” in the middle of the screen in the Android Phone.
2. Create an application with login module. (Check username and password).
3. Create spinner with strings taken from resource folder (res >> value folder) and on changing the spinner value, Image will change.
4. Create a menu with 5 options and selected option should appear in text box.
5. Create a list of all courses in your college and on selecting a particular course teacher-in-charge of that course should appear at the bottom of the screen.
6. Create an application with three option buttons, on selecting a button colour of the screen will change.
7. Create and Login application as above. On successful login, pop up the message.
8. Create an application to Create, Insert, update, Delete and retrieve operation on the database.

---

**SEC – 2**

**Total Marks: 75**

**Practical: 50**

**Internal Assessment: 25**

**2 Lectures , 2 Practicals ( each in group of 15 to 20)**

### **PHP Programming**

**(1 +2 Lab)**

#### **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**

HTML

**(5L)**

- 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**

**(2L)**

- 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

### **Software Lab Based on PHP**

1. Create a PHP page using functions for comparing three integers and print the largest number.
2. Write a function to calculate the factorial of a number (non-negative integer). The function accept the number as an argument.
3. WAP to check whether the given number is prime or not.
4. Create a PHP page which accepts string from user. After submission that page displays the reverse of provided string.
5. Write a PHP function that checks if a string is all lower case.
6. Write a PHP script that checks whether a passed string is palindrome or not? ( A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run)
7. WAP to sort an array.
8. Write a PHP script that removes the whitespaces from a string.  
Sample string : 'The quick " " brown fox'  
Expected Output : Thequick""brownfox
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.