

**PROPOSED SYLLABUS FOR CHOICE BASED CREDIT SYSTEM IN**  
**BACHELOR OF COMPUTER APPLICATION (BCA)**  
**Department of Computer Science**  
**North Lakhimpur College (Autonomous)**

**Contents**

Core Papers : 14 (Credit: 06 each).....	8
C-101: Programming Fundamentals using C/C++.....	8
LAB (C-101): Programming Fundamentals using C/C++ Lab.....	9
C-102: Computer System Architecture.....	11
LAB (C-102): Computer System Architecture Lab.....	12
C-201: Operating Systems.....	13
LAB (C-201): Operating Systems Lab.....	14
C-202: Data Structures.....	<b>Error! Bookmark not defined.</b>
LAB (C-202): Data Structures Lab.....	16
C-301: Discrete Structures.....	17
Tutorial (C-301): Discrete Structures Tutorial.....	17
C-302: Internet Technologies.....	18
LAB (C-302): Internet Technologies Lab.....	18
C-303: Theory of Computation.....	18
Tutorial (C-303): Theory of Computation.....	19
C-401: Design and Analysis of Algorithms.....	19
LAB (C-401): Design and Analysis of Algorithms Lab.....	20
C-402: Programming in Java.....	21
LAB (C-402): Programming in Java Lab.....	22
C-403: Database Management Systems.....	23
LAB (C-403): Database Management Systems Lab.....	24
C-501: Software Engineering.....	26
LAB (C-501): Software Engineering Tutorial.....	27
C-502: Computer Networks.....	27
LAB (C-502): Computer Networks Lab.....	27
C-601: Artificial Intelligence.....	28
LAB (C-601): Artificial Intelligence Lab.....	28
C-602: Computer Graphics.....	29
LAB (C-602): Computer Graphics Lab.....	29
Foundation Course: 04 (Credit: 06 each).....	31
FC - 101: Mathematics -I.....	31
Tutorial : Mathematics I.....	31
FC – 201: Mathematics -II.....	<b>Error! Bookmark not defined.</b>

Tutorial : Mathematics II.....	<b>Error! Bookmark not defined.</b>
FC – 301: Accounting and Financial Management.....	33
LAB (FC – 301): Accounting and Financial Management.....	33
FC - 401: Probability and Statistics.....	33
Tutorial : Probability and Statistics.....	34
Skill Enhancement Course : (Credit: 02 each).....	35
SEC – 301 (a) PHP Programming (1 +2 Lab).....	35
Software Lab Based on PHP:.....	36
SEC – 301 (b) XML Programming.....	36
Software Lab Based on XML:.....	37
SEC – 401 (a) Android Programming:.....	38
Software Lab Based on Android Programming:.....	39
SEC – 401 (b) Programming in Python.....	39
Software Lab Based on Python.....	40
Discipline Specific Elective Papers : (Credit: 06 each).....	42
DSE – 501 (a) Numerical Methods.....	42
Numerical Methods Lab.....	42
DSE – 501 (b) Combinatorial Optimization.....	43
Tutorial : Combinatorial Optimization.....	43
DSE – 501 (c) Microprocessor.....	43
Microprocessor Lab.....	44
DSE – 502 (a) Information Security.....	44
Information Security Lab.....	44
DSE – 502 (b) Digital Image Processing.....	45
Digital Image Processing Lab.....	46
DSE – 502 (c) Data Mining.....	47
Data Mining Lab.....	47
DSE – 601 (a) Systems Programming.....	47
Systems Programming LAB.....	48
DSE – 601 (b) Introduction to Data Science.....	48
Introduction to Data Science Lab.....	49
DSE – 601 (c) Cloud Computing.....	50
Cloud Computing Lab.....	50
DSE – 602 (a) Dissertation / Project work.....	51
DSE – 602 (b) Network Programming.....	51
Network Programming Lab.....	51
DSE – 602 (c) Machine Learning.....	51
Machine Learning Lab.....	52

PROPOSED SCHEME FOR CHOICE BASED CREDIT SYSTEM IN

**BACHELOR OF COMPUTER APPLICATION (BCA)**

**Department of Computer Science**

**North Lakhimpur College (Autonomous)**

Semester	Core Course	Ability Enhancement Compulsory Course (AECC)	Skill Enhancement Course (SEC)	Discipline Specific Elective (DSE)	Foundation Course (FC)
<b>No. of Courses</b>	<b>14</b>	<b>2</b>	<b>2</b>	<b>4</b>	<b>4</b>
<b>I</b>	Programming Fundamentals using C/C++	English			Mathematics-I
	Computer System Architecture				
<b>II</b>	Operating Systems	Environmental Science			Mathematics-II
	Data Structures				
<b>III</b>	Discrete Structures		SEC-1		Accounting and Financial Management
	Internet Technologies				
	Theory of Computation				
<b>IV</b>	Design and Analysis of Algorithms		SEC-2		Probability and Statistics
	Programming in JAVA				
	Database Management Systems				
<b>V</b>	Software Engineering			DSE-1	
	Computer Networks			DSE-2	
<b>VI</b>	Artificial Intelligence			DSE-3	
	Computer Graphics			DSE-4	

Semester	Course Opted	Course Name	Credits
<b>I</b>	Ability Enhancement Compulsory Course-I	English	2
	Core course-I	Programming Fundamentals using C/C++	4
	Core course-I Practical	Programming Fundamentals using C/C++ LAB	2
	Core course-II	Computer System Architecture	4
	Core course-II Practical	Computer System Architecture LAB	2
	Foundation Course –I	Mathematics –I	5
	Foundation Course –I Tutorial	Mathematics – I Tutorial	1
<b>II</b>	Ability Enhancement Compulsory Course-I	Environmental Science	2
	Core course-III	Operating Systems	4
	Core course-III Practical	Operating Systems LAB	2
	Core course-IV	Data Structures	4
	Core course-IV Tutorial	Data Structures LAB	2
	Foundation Course –II	Mathematics-II	5
	Foundation Course –II Tutorial	Mathematics-II Tutorial	1
<b>III</b>	Core course-V	Discrete Structures	5
	Core course-V Practical	Discrete Structures Tutorial	1
	Core course-VI	Internet Technologies	4
	Core course-VI Practical	Internet Technologies LAB	2
	Core course-VII	Theory of Computation	5
	Core course-VII Practical	Theory of Computation Tutorial	1
	Skill Enhancement Course-I	SEC-I	2
	Foundation Course –III	Accounting and Financial Management	4
	Foundation Course –III Tutorial	Accounting and Financial Management LAB	2

<b>IV</b>	Core course-VIII	Design and Analysis of Algorithms	4
	Core course-VIII Practical	Design and Analysis of Algorithms LAB	2
	Core course-IX	Programming in JAVA	4
	Core course-IX Practical	Programming in JAVA LAB	2
	Core course-X	Database Management Systems	4
	Core course-X Practical	Database Management Systems LAB	2
	Skill Enhancement Course-II	SEC-II	2
	Foundation Course –IV	Probability and Statistics	5
	Foundation Course –III Tutorial	Probability and Statistics Tutorial	1
<b>V</b>	Core course-XI	Software Engineering	5
	Core course-XI Practical	Software Engineering Tutorial	1
	Core course-XII	Computer Networks	4
	Core course-XII Tutorial	Computer Networks LAB	2
	Discipline Specific Elective – I	DSE-I	4
	Discipline Specific Elective – I Practical	DSE-I LAB	2
	Discipline Specific Elective – II	DSE-II	4
	Discipline Specific Elective – II Practical	DSE-II LAB	2
<b>VI</b>	Core course-XIII	Artificial Intelligence	4
	Core course-XIII Practical	Artificial Intelligence LAB	2
	Core course-XIV	Computer Graphics	4
	Core course-XIV Practical	Computer Graphics LAB	2
	Discipline Specific Elective – III	DSE-III	4
	Discipline Specific Elective – III Practical	DSE-III LAB	2
	Discipline Specific Elective – IV	DSE-IV	4
	Discipline Specific Elective – IV Practical	DSE-IV LAB	2
<b>Total Credits</b>			<b>140</b>



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

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

**Foundation Course: (Credit: 06 each)**

- (a) Mathematics I
- (b) Mathematics II
- (c) Accounting & Financial Management
- (d) Probability and Statistics

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

**DSE – 1 (any one)**

- (a) Numerical Methods
- (b) Combinatorial Optimization
- (c) Microprocessor

**DSE – 2 (any one)**

- (a) Information Security
- (b) Digital Image Processing
- (c) Data Mining

**DSE – 3 (any one)**

- (a) System Programming
- (b) Introduction to Data Science
- (c) Cloud Computing

**DSE – 4 (any one)**

- (a) Project Work / Dissertation
- (b) Network Programming
- (c) Machine Learning

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

**SEC – 1 (any one)**

- (a) PHP Programming
- (b) XML Programming

**SEC – 2 (any one)**

- (a) Android Programming
- (b) Programming in Python

## Core Papers : 14 (Credit: 06 each)

### C-101: Programming Fundamentals using C/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**

**(8 Lectures)**

Inheritance, Polymorphism (Virtual Functions, Pure Virtual Functions).

### **Reference Books**

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

### **LAB (C-101): Programming Fundamentals using C/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 perform following actions on an array entered by the user:

- a. Print the even-valued elements
- b. Print the odd-valued elements
- c. Calculate and print the sum and average of the elements of array
- d. Print the maximum and minimum element of array
- e. Remove the duplicates from the array
- f. 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.

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

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

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

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

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

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

22. 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

23. 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).

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

25. 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.
26. Create a structure Student containing fields for Roll No., Name, Class, Year and Total Marks. Create 10 students and store them in a file.
27. Write a program to retrieve the student information from file created in previous question and print it in following format: *Roll No. Name Marks*
28. Copy the contents of one text file to another file, after removing all whitespaces.
29. Write a function that reverses the elements of an array in place. The function must accept only one pointer value and return void.
30. 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.

## C-102: Computer System Architecture

Theory: 60 Lectures

### 1. 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

### 2. Digital Logic (8 lectures)

Boolean algebra, Logic gates, SOP, POS, combinational circuits, logic circuit design using NAND and NOR gates, circuit simplification using K-Map upto four variables, decoders, multiplexers, sequential circuits, flip-flops, registers, counters and memory units.

### 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.

### 5. Memory Organization (6 lectures)

Cache memory, Associative memory, mapping.

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

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

## Reference Books:

- M. Mano, Computer System Architecture, Pearson Education 1992
- J. Dos Reis, Assembly Language and Computer Architecture using C++ and JAVA, Course Technology, 2004
- W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition, Prentice Hall of India, 2009

## LAB (C-102): Computer System Architecture Lab

Practical: 60 Lectures

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

a. CLA	e. CIR	i. SNA
b. CLE	f. CIL	j. SZA
c. CMA	g. INC	k. SZE
d. CME	h. SPA	l. HLT
4. Simulate the machine for the following memory-reference instructions with I= 0 and address part = 082. The instruction to be stored at address 022 in RAM. Initialize the memory word at address 082 with the operand B8F2 and AC with A937. Determine the contents of AC, DR, PC, AR and IR in hexadecimal after the execution.
5. 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 Application**  
**Semester – II**  
**Core Course                      BCA-CC-T-201**

**Operating Systems**

**70 (Th-56+IA-14)**

**Credit : 4**

**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.

**Reference Books:**

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

## LAB (C-201): 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 a program to implement first-fit, best-fit and worst-fit allocation strategies.

# **Computer Application**

## **Semester – II**

### **Core Course BCA-CC-T-202**

#### **Data Structures**

**70 (Th-56+IA 14**

**Credit : 4**

#### **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, Shell Sort, Comparison of Sorting Techniques

##### **8. Hashing (5 Lectures)**

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

#### **Reference Books:**

- Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.
- SartajSahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
- Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using C and C++", Second edition, PHI, 2009.
- Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011

- Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub, 2003
- John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 edition, 2009
- Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley, 2013
- Herbert Schildt, "Java The Complete Reference (English) 9th Edition Paperback", Tata McGraw Hill, 2014.
- D. S. Malik, P.S. Nair, "Data Structures Using Java", Course Technology, 2003.

## **LAB (C-202): 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. 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.

## **C-301: Discrete Structures**

**Theory: 60 Lectures**

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

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

### **2. Algebraic Structure: (18 Lectures)**

Operations, semi groups, Groups, Sub-groups, Normal Sub-groups & Homomorphism.

Rings, Integral Domains & Fields – **fundamental concepts only**

Vector spaces and subspaces

### **3. Graph Theory: (15 Lectures)**

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

### **4. Propositional Logic (12 Lectures)**

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

### **Reference Books:**

- C.L. Liu , D.P. Mohapatra, Elements of Discrete mathematics, 2nd Edition , Tata McGraw Hill, 1985,
- Kenneth Rosen, Discrete Mathematics and Its Applications, Sixth Edition ,McGraw Hill 2006
- T.H. Cormen, C.E. Leiserson, R. L. Rivest, Introduction to algorithms, 3rd edition Prentice Hall on India, 2009
- M. O. Albertson and J. P. Hutchinson, Discrete Mathematics with Algorithms , John wiley Publication, 1988
- J. L. Hein, Discrete Structures, Logic, and Computability, 3rd Edition, Jones and Bartlett Publishers, 2009
- D.J. Hunter, Essentials of Discrete Mathematics, Jones and Bartlett Publishers, 2008

## **Tutorial (C-301): Discrete Structures Tutorial**

**Tutorial: 15 lectures**

## **C-302: Internet Technologies**

**Theory: 60 Lectures**

### **1. Java**

**(5 lectures)**

Use of Objects, Array and Array List class

### **2. JavaScript**

**(15 lectures)**

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

### **3. JDBC**

**(10 lectures)**

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

### **4. JSP**

**(20 lectures)**

Introduction to 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.

### **5. Java Beans**

**(10 lectures)**

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

### **Reference Books:**

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

## **LAB (C-302): 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. Find the sum and average of n 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.

## **C-303: Theory of Computation**

**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, Deterministic and non-deterministic finite automata, NFA to DFA Conversion, Regular languages and their relationship with finite automata, Pumping lemma and closure properties of regular languages.

### **3. Context free languages**

**(17 Lectures)**

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

### **4. Turing Machines and Models of Computations**

**(15 Lectures)**

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

#### **Reference Books:**

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

### **Tutorial (C-303): Theory of Computation**

**Tutorial: 15 Lectures**

### **C-401: Design and Analysis of Algorithms**

**Theory: 60 Lectures**

#### **1. Introduction**

**(10 Lectures)**

Basic Design and Analysis techniques of Algorithms, Correctness of Algorithm, Asymptotic notation

Recurrence Relations, generating functions, Linear Recurrence Relations with constant coefficients and their solution, Substitution Method, Recurrence Trees, Master Theorem Master Theorem (without proof)

#### **2. Algorithm Design Techniques**

**( 6 Lectures)**

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

#### **3. Sorting and Searching Techniques**

**(17 Lectures)**

Elementary sorting techniques–Bubble Sort, Insertion Sort, Merge Sort, Advanced Sorting techniques - Heap Sort, Quick Sort, Sorting in Linear Time - Bucket Sort, Radix Sort and Count Sort, Searching Techniques, Medians & Order Statistics, complexity analysis;

#### **4. Lower Bounding Techniques**

**(5 Lectures)**

Decision Trees

#### **5. Balanced Trees**

**(7 Lectures)**

Red-Black Trees

#### **6. Graphs**

**10 Lectures)**

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

#### **7. String Processing(5 Lectures)**

String Matching, KMP Technique

#### **Reference Books:**

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

## LAB (C-401): Design and Analysis of Algorithms Lab

### Practical: 60 Lectures

1. Implement Insertion Sort & 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  $n \log n$ .*

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

## **C-402: Programming in Java**

**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**

- Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
- 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.
- Joshua Bloch, "Effective Java" 2nd Edition, Publisher: Addison-Wesley, 2008.

- Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 1 ,9th Edition,Printice Hall.2012
- Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, Printice Hall.2013
- Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
- E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
- Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011.
- "Head First Java", Orielly Media Inc. 2nd Edition, 2005.
- David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.
- John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition, 2004.

### **LAB (C-402): Programming in Java Lab**

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

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of .length in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument
8. Write a program that show working of different functions of String and 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 URLConnection 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.

## **C-403: Database Management Systems**

**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.

### 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+ trees.

#### Reference Books :

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

#### LAB (C-403): 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 Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
31. Query to display the Employee No. And Name for all employees who earn more than the average salary.
32. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a \_T'.
33. Query to display the names and salaries of all employees who report to King.
34. Query to display the department no, name and job for all employees in the Sales department.

## **C-501: Software Engineering**

**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, COCOMO

### **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:**

- R.S. Pressman, Software Engineering: A Practitioner's Approach (7th Edition), McGraw-Hill, 2009.
- P. Jalote, An Integrated Approach to Software Engineering (2nd Edition), Narosa Publishing House, 2003.

- K.K. Aggarwal and Y. Singh, Software Engineering ( 2nd Edition), New Age International Publishers, 2008.
- Sommerville, Software Engineering (8th edition), Addison Wesley, 2006.
- D. Bell, Software Engineering for Students (4th Edition), Addison-Wesley, 2005.
- R. Mall, Fundamentals of Software Engineering (2nd Edition), Prentice-Hall of India, 2004.

## **LAB (C-501): Software Engineering Tutorial**

**Tutorial: 15 lectures**

## **C-502: Computer Networks**

**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 and 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, **IP addressing**

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

Transport services- error and flow control, congestion control, Connection establishment and release- three way handshakes, **port**

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

Overview of DNS protocol; overview of WWW &HTTP protocol, telnet, ftp.

## **Reference Books**

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

## **LAB (C-502): 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.

## **C-601: Artificial Intelligence**

### **Theory: 60 Lectures**

#### **1. Introduction**

**(6 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**

**(8 Lectures)**

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

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

**(6 Lectures)**

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

### **Reference Books:**

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

## **LAB (C-601): 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.

## **C-602: Computer Graphics**

**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.

### **Reference Books:**

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

## **LAB (C-602): Computer Graphics Lab**

**Practical: 60 Lectures**

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

## Foundation Course: 04 (Credit: 06 each)

### FC - 101: Mathematics -I

**1. Matrices :** Elementary concept of matrix and determinants, Types, Matrix Arithmetic, Boolean Matrix, Solution of Linear Equation by Matrix Method, Rank of Matrix, Eigen Values and Eigen Vectors.

**2. Number Theory:** Basic Properties, Divisibility Theory, Congruence, Applications of Congruence

**3. Posets and Lattices:** Partially order Sets, Lattice, Lattice as algebraic system, SubLattice, Finite Boolean Algebra.

**4. Complex Numbers:**

Complex number as an ordered pair, operations on complex numbers, De-Moivre's Theorem, roots of complex numbers

**5. Elements of Coding Theory**

Group codes, Parity-Check and Generator Matrix, Hamming Codes

#### Books:

1. Kreyszig E. “*Advanced Engineering Mathematics*”, Tenth Edition, Wiley, 2010.
2. Swapan Kr. Sarkar, A Textbook of Discrete Mathematics, S. Chand
3. Tremblay Manohar, Discrete Mathematical Structures with Application to Computer Science, Tata McGraw Hill
4. Garg R.L., Gupta N., “*Engineering Mathematics*”, 1st Edition, Pearson, 2015.

### Tutorial : Mathematics I

**Tutorial: 15 Lectures**

**Computer Application**  
**Semester – II**  
**Foundation Course - BCA-FC-T4-201**

**Mathematics -II**

**70 (Th-56+IA 14**

**Credit : 4**

**Unit I – Introduction to Differential Calculus:**

Limits, continuity and differentiability, ordinary differentiation, partial differentiation, indeterminate forms. Rolle's Theorem, MVTs

**Unit II-Expansion of functions:**

Taylor's and Maclaurin's theorems, Euler's theorem on homogeneous functions.

**Unit III-Maxima and Minima:**

Maxima and minima of functions of single variable and two variables.

**Unit IV - Integral Calculus:**

Indefinite integral, definite integrals, reduction formulae, application of integral calculus – length, area, volume. Idea of multiple integrals.

**Unit V - Transform Calculus:**

Laplace Transforms, Inverse Laplace Transform.

**Books:**

1. Kreyszig E. "*Advanced Engineering Mathematics*", Tenth Edition, Wiley, 2010.
2. Swapan Kr. Sarkar, A Textbook of Discrete Mathematics, S. Chand
3. Tremblay Manohar, Discrete Mathematical Structures with Application to Computer Science, Tata McGraw Hill
4. Ayres F., Mendelson E. "*Schaum's Outline of Calculus*", 6th Edition, McGraw Hill Education, 2013.
5. Silverman R.A., "*Essential Calculus with Applications*", 5<sup>th</sup> Edition, Dover Publications, 2014.
6. Garg R.L., Gupta N., "*Engineering Mathematics*", 1st Edition, Pearson, 2015.



## FC – 301: Accounting and Financial Management

### Unit-1:

Meaning and definition of accounting, parties or users interested in accounting, branches of accounting. Accounting concepts and conventions. Basic accounting terminologies, Classification of accounts, Journal entry, ledger posting and balancing of ledger. Subsidiary books – meaning and importance, preparation of cash book (Triple Column).

### Unit-2:

Trial Balance-concept, objectives: Financial statements-meaning, objectives, preparation of Trading and Profit and Loss Accounts, Balance Sheet of sole trading concern. Classification of Assets and Liabilities. Depreciation- meaning, causes, accounting for depreciation. Accounting software-Tally (introductory part).

### Unit-3:

Financial Management-meaning and objectives, functions of financial management. Concept of capital structure-computation of Cost of Capital; Management of Working capital-need of working capital, operating cycle, sources of working capital.

### Unit-4:

Budget and Budgetary Control-definition, objectives of budget, classification, advantage, characteristics of budget. Preparation of production/sales and cash budget. Capital Budgeting-meaning, importance and methods of capital budgeting. Concept of Marginal costing; Cost-Volume-Profit analysis, Break-even Point.

1. B.B.Dam; R.A.Sarda; R.Barman; B.Kalita: “*Theory and Practice of Accountancy (V-I)*”, Capital Publishing Company, Guwahati.
2. C.M.Juneja; R.C.Chowla; K.K.Saxena; “*Book-Keeping and Accountancy (V-I)*”, Kalyani Publishers, Ludhiana..
3. R.K.Sharma; S.K.Gupta: “*Management Accounting*”. Kalyani Publishers, Ludhiana.
4. M.Y. Khan; P.K.Jain: “*Principles of Financial Management*”. Tata McGraw Hills, New Delhi

## LAB (FC – 301): Accounting and Financial Management

Practical implementation using TALLY or similar Financial accounting packages

## FC - 401: Probability and Statistics

1. **Introduction and Overview:** Concepts of data, data collection methods, the distinction between populations and samples and between population parameters and sample statistics; the use of measures of location and variation to describe and summarize data; population moments and their sample counterparts. Measures of central tendency, dispersion
2. **Elementary Probability Theory:** Sample spaces and events; probability axioms and properties; counting techniques; conditional probability and Bayes’ rule; independence.
3. **Random Variables and Probability Functions:** Density and distribution functions for jointly distributed random variables; computing expected values; covariance.

- 4. Probability Distributions:** Defining random variables; probability distributions; expected values of random variables and of functions of random variables; properties of commonly used discrete and continuous distributions (uniform, binomial, normal, poisson and exponential random variables).
- 5. Correlation and regression:** introduction to correlation, Karl-Pearson correlation coefficient, idea of rank correlation, method of least square, basic idea of regression, linear regression

Readings:

1. Jay L. Devore, Probability and Statistics for Engineers, Cengage Learning, 2010.
2. John E. Freund, Mathematical Statistics, Prentice Hall, 1992.
3. Richard J. Larsen and Morris L. Marx, An Introduction to Mathematical Statistics and its Applications, Prentice Hall, 2011.
4. William G. Cochran, Sampling Techniques, John Wiley, 2007.

## **Tutorial : Probability and Statistics**

**Tutorial: 15 Lectures**

## Skill Enhancement Course : (Credit: 02 each)

### SEC – 301 (a) PHP Programming (1 +2 Lab)

#### 1.Introduction to PHP: (2Lectures)

- 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

#### 2. Handling HTML form with PHP: (2Lectures)

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

#### 3. PHP conditional events and Loops: (3Lectures)

- PHP IF Else conditional statements ( Nested IF and Else)
- Switch case, while ,For and Do While Loop
- Goto , Break ,Continue and exit

#### 4. PHP Functions: (3Lectures)

- 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

#### 5. String Manipulation and Regular Expression: (3Lectures)

- 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

#### 6. Array: (3Lectures)

- 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

#### 7. Database Connectivity: (1Lectures)

- Connection with MySql

#### Reference Books:

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

- Luke Welling, Laura Thompson, "PHP and MySQL Web Development", 4th Edition, Addition Paperback, Addison-Wesley Professional, 2008.
- David Sklar, Adam Trachtenberg, "PHP Cookbook: Solutions & Examples for PHP Programmers", 2014.

## 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. Write a simple PHP program to check that emails are valid.
14. Insert and retrieve data from MySql using PHP
15. WAP to print first n even numbers.
16. \$color = array('white', 'green', 'red') Write a PHP script which will display the colors in the following way : Output : white, green, red, • green • red • white
17. Using switch case and dropdown list display a —Hello! message depending on the language selected in drop down list.
18. Write a PHP program to print Fibonacci series using recursion.
19. Write a PHP script to replace the first 'the' of the following string with 'That'. Sample : 'the quick brown fox jumps over the lazy dog.' Expected Result : That quick brown fox jumps over the lazy dog.

## SEC – 301 (b) XML Programming

Theory classes : 15 lectures

- |   |                     |
|---|---------------------|
| <b>1. Introduction:</b>   | <b>(3 lectures)</b> |
| Understanding Mark-up Languages, Introduction to XML and its Goals. |                     |
| <b>2. XML Basics:</b>   | <b>(5Lectures)</b>  |
| XML Structure and Syntax, Document classes and Rules.               |                     |
| <b>3. Other XML Concepts:</b>                                       | <b>(4 Lectures)</b> |

#### 4. XML with Style:

(3 lectures)

XSL –Style Sheet Basics, XSL basics, XSL style sheets.

##### Books Recommended

- XML in action web technology by William J. Pardi
- Step by Step XML by Michael J. Young

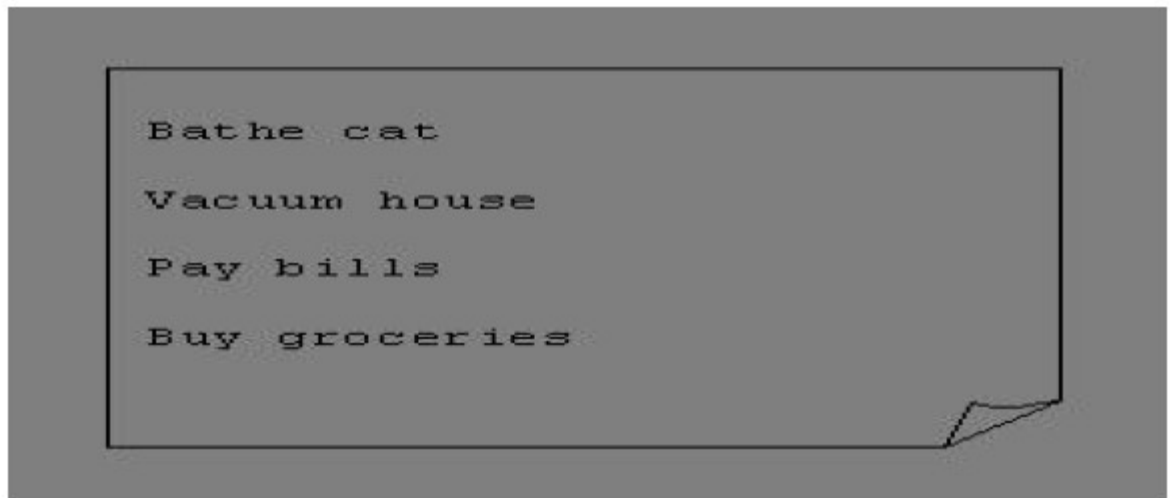
##### Software Lab Based on XML:

###### Exercise #1 – Information Structure

In this exercise, student will practice identifying the structure of an information object. For the sample document provided below:

Label the information structures you see, including containing structures.

1. Draw a tree representation of the structure.



2. Exercise 2# Deconstructing an XML Document

In this exercise, student will practice identifying the explicit structure within an XML document. In a sense, this is the reverse of what you did in Exercise #1. For the sample XML markup below, create a document-like representation (or a simple drawing) for the content contained within the XML tags:

```
<book>
<coverInfo>
<title>The XML Handbook</title>
<author>Charles F. Goldfarb</author>
<author>Paul Prescod</author>
<edition>Second</edition>
<description>The definitive XML resource: applications, products, and technologies.
Revised and expanded—over 600 new pages. </description>
</coverInfo>
</book>
```

3. Exercise #3 – Creating XML Markup

In this exercise, create some XML markup based on the tree representation from Exercise #1 above, and the content from the original sample document.

4. Exercise #4 – Well-Formedness

This exercise checks your understanding of the constraints for well-formedness. Are the following document instances well-formed? Explain any **NO** answers.

```

<list><title>The first list</title><item>An item</item>
<item>An item</item><item>Another item</item>
<para>Bathing a cat is a <emph>relatively</emph> easy task as long as the cat is
willing.</para>
<bibl><title>How to Bathe a Cat<author></title>Merlin Bauer<author></bibl>

```

##### 5. Exercise #5-Well Formedness

This exercise is a bit more challenging than the previous example. Here is a fragment of an XML document instance. Identify all the places where it fails to match the constraints for well formedness.

```

<PROCEDURE>
<TITLE>How to Bathe a Cat</TITLE>
<OVERVIEW> This procedure tells you how to bathe a cat.
<WARNING></OVERVIEW>
Cats don't like to take baths. You could get hurt doing this. Be sure to obtain all the
required protective gear before you start. </WARNING>
<EQUIPEMENT>
<ITEM>Hockey Mask</ITEM>
<ITEM>Padded Full-body Kevlar Armor</ITEM>
<ITEM>Tub full of warm water</ITEM>
<ITEM>Towels </ITEM>
<ITEM>FirstAidkit</ITEM>
<ITEM>Cat Shampoo</ITEM>
<EQUIPMENT>
<INSTRUCTIONS><STEP> Locate the cat, who by now is hiding under the
bed.</STEP>
<STEP>Place the cat in the tub of water.</STEP><ITEM>Using the First Aid kit, repair
the damage to your head and arms.</STEP><STEP>Place the cat back in the tub and hold
it down.</STEP>
<STEP>Wash it really fast, then make an effort to dry it with the towels.</STEP>
<STEP>Decide not to do this again. </STEP>
</INSTRUCTIONS>

```

**Note: Cover more exercises based on XML Programming theory concepts.**

## SEC – 401 (a) Android Programming:

### 1. Introduction: (2 lectures)

History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture.

### 2. Overview of object oriented programming using Java: (4lectures)

OOPs Concepts: Inheritance, Polymorphism, Interfaces, Abstract class, Threads, Overloading and Overriding, Java Virtual Machine.

### 3. Development Tools: (5 lectures)

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.

### 4. User Interface Architecture: (2 lectures)

Application context, intents, Activity life cycle, multiple screen sizes

### 5. User Interface Design: (2 lectures)

Form widgets, Text Fields, Layouts, Button control, toggle buttons, Spinners(Combo boxes), Images, Menu, Dialog.

## **6. Database:**

**(2 lectures)**

Understanding of SQLite database, connecting with the database.

### **Reference Books:**

- 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 – 401 (b) Programming in Python**

### **1. Planning the Computer Program:**

**(2 Lectures)**

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

### **2. Techniques of Problem Solving:**

**(2 Lectures)**

Flowcharting, decision table, algorithms, Structured programming concepts, Programming methodologies viz. top-down and bottom-up programming.

### **3. Overview of Programming :**

**(3 Lectures)**

**4. Introduction to Python:**

**(4 Lectures)**

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

**5. Creating Python Programs :**

**(4 Lectures)**

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

**Reference Books**

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

**Software Lab Based on Python:**

**Section: A ( Simple programs)**

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

**Section: B (Visual Python):**

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

1. Write a menu-driven program to create mathematical 3D objects  
I. curve



II. sphere

III. cone

IV. arrow

V. ring

VI. cylinder.

2. WAP to read n integers and display them as a histogram.

3. WAP to display sine, cosine, polynomial and exponential curves.

4. WAP to plot a graph of people with pulse rate p vs. height h. The values of p and h are to be entered by the user.

5. WAP to calculate the mass m in a chemical reaction. The mass m (in gms) disintegrates according to the formula  $m=60/(t+2)$ , where t is the time in hours. Sketch a graph for t vs. m, where  $t \geq 0$ .

6. A population of 1000 bacteria is introduced into a nutrient medium. The population p grows as follows:

$$P(t) = (15000(1+t))/(15 + e^t)$$

where the time t is measured in hours. WAP to determine the size of the population at given time t and plot a graph for P vs t for the specified time interval.

7. Input initial velocity and acceleration, and plot the following graphs depicting equations of motion:

I. velocity wrt time ( $v=u+at$ )

distance wrt time ( $s=ut+0.5at^2$ )

distance wrt velocity ( $s=(v^2-u^2)/2a$ )

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

**DSE – 501 (a) Numerical Methods**

**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, Bisection method, Secant method, Regula-Falsi method, Newton-Raphson method, Newton's method for solving nonlinear systems, Gauss elimination method (with row pivoting) and Gauss-Jordan method, Gauss Thomas method for tridiagonal systems, **Iterative methods:** Jacobi and Gauss-Seidel iterative methods

**Interpolation:**

Lagrange's form and Newton's form

Finite difference operators, Gregory Newton forward and backward differences Interpolation

**Piecewise polynomial interpolation:** Linear interpolation, Cubic spline interpolation (only method),

**Numerical differentiation:** First derivatives and second order derivatives, Richardson extrapolation

**Numerical integration:** Trapezoid rule, Simpson's rule (only method), Newton-Cotes open formulas

**Extrapolation methods:** Romberg integration, Gaussian quadrature, Ordinary differential equation: Euler's method

**Modified Euler's methods:** Heun method and Mid-point method,

**Runge-Kutta second methods:** Heun method without iteration, Mid-point method and Ralston's method

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

**Reference Books:**

- Laurence V. Fausett, Applied Numerical Analysis, Using MATLAB, Pearson, 2/e (2012)
- 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)
- Steven C Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists, Tata McGraw Hill, 2/e (2010)

**Numerical Methods Lab**

**Practical: 60 lectures**

1. Find the roots of the equation by bisection method.
2. Find the roots of the equation by secant/Regula-Falsi method.
3. Find the roots of the equation by Newton's method.
4. Find the solution of a system of nonlinear equation using Newton's method.
5. Find the solution of tridiagonal system using Gauss Thomas method.
6. Find the solution of system of equations using Jacobi/Gauss-Seidel method.
7. Find the cubic spline interpolating function.

8. Evaluate the approximate value of finite integrals using Gaussian/Romberg integration.
9. Solve the boundary value problem using finite difference method.

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

## **DSE – 501 (b)Combinatorial Optimization**

### **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

### **Transportation and Assignment problem 10 hrs**

Transportation Model, NWC, LCM, VAM, MODI method, Hungarian method for solving assignment 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: Combinatorial Optimization**

**Tutorial: 15 Lectures**

## **DSE – 501 (c)Microprocessor**

**Theory: 60 Lectures**

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

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

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

### **Reference Books:**

- Barry B. Brey : The Intel Microprocessors : Architecture, Programming and Interfacing. Pearson Education, Sixth Edition, 2009.
- 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

## DSE – 502 (a) Information Security

### Theory: 60 Lectures

#### 1. Introduction

Security, Attacks, Computer Criminals, Security Services, Security Mechanisms.

#### 2. Cryptography

Substitution ciphers, Transpositions Cipher, Confusion, diffusion, Symmetric, Asymmetric Encryption. DES Modes of DES, Uses of Encryption, Hash function, key exchange, Digital Signatures, Digital Certificates.

#### 3. Program Security

Secure programs, Non malicious Program errors, Malicious codes virus, Trap doors, Salami attacks, Covert channels, Control against program

#### 4. Threats.

Protection in OS: Memory and Address Protection, Access control, File Protection, User Authentication.

#### 5. Database Security

Requirements, Reliability, Integrity, Sensitive data, Inference, Multilevel Security.

#### 6. Security in Networks

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

#### 7. Administrating Security

Security Planning, Risk Analysis, Organisational Security Policy, Physical Security. Ethical issues in Security: Protecting Programs and data. Information and law.

#### Reference Books:

- C. P. Pfleeger, S. L. Pfleeger; Security in Computing, Prentice Hall of India, 2006
- W. Stallings; Network Security Essentials: Applications and Standards, 4/E, 2010

## Information Security Lab

### Practical: 60 lectures

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois

2. Use of Password cracking tools : John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations.
4. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these operations.
5. Use nmap/zenmap to analyse a remote machine.
6. Use Burp proxy to capture and modify the message.
7. Demonstrate sending of a protected word document.
8. Demonstrate sending of a digitally signed document.
9. Demonstrate sending of a protected worksheet.
10. Demonstrate use of steganography tools.
11. Demonstrate use of gpg utility for signing and encrypting purposes.

## **DSE – 502 (b) Digital Image Processing**

**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

- R C Gonzalez , R E Woods, Digital Image Processing, 3rd Edition, Pearson Education.2008.
- A K Jain, Fundamentals of Digital image Processing, Prentice Hall of India.1989.
- K R Castleman, Digital Image Processing, Pearson Education.1996
- Schalkoff, Digital Image Processing and Computer Vision, John Wiley and Sons.1989.
- 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 a program in C and MATLAB/SCILAB for edge detection using different edge detection mask
  10. Write and execute program for image morphological operations erosion and dilation.
  11. To write and execute program for wavelet transform on given image and perform inverse wavelet transform to reconstruct image.

## **DSE – 502 (c) Data Mining**

### **Introduction :** **(6Lectures)**

Basic Data Mining Tasks, Data Mining Issues, Data Mining Metrics, Data Mining from a Database Perspective.

### **Data Mining Techniques :** **(10Lectures)**

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

### **Classification :** **(15Lectures)**

Statistical-Based Algorithms, Distance-Based Algorithms, Decision Tree-Based Algorithms, Neural Network-Based Algorithms, Rule-Based Algorithms, Combining Techniques.

### **Clustering :** **(12Lectures)**

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

### **Association Rules :** **(12Lectures)**

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

### **Advanced Techniques :** **(5Lectures)**

Web Mining, Spatial Mining, Temporal Mining.

## **Data Mining Lab**

### **Practical: 60 lectures**

Practical exercises based on concepts listed in theory.

## **DSE – 601 (a) Systems Programming**

### **Theory: 60 lectures**

#### **1. Assemblers& Loaders, Linkers:** **(10 lectures)**

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

#### **2. Introduction:** **(2 lectures)**

Overview of compilation, Phases of a compiler

**3. Lexical Analysis:** (6 lectures)

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

**4. Parsing:** (10 lectures)

Bottom up parsing- LR parser, yacc.

**5. Intermediate representations** (10 lectures)

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

**6. Storage organization:** (5 lectures)

Activation records, stack allocation

**7. Code Generation:** (5 lectures)

Object code generation

**Reference Books**

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

## Systems Programming LAB

**Practical: 60 lectures**

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.

## DSE – 601 (b) Introduction to Data Science

**Theory: 60 Lectures**

**1. Data Scientist's Tool Box:**

Turning data into actionable knowledge, introduction to the tools that will be used in building data analysis software: version control, markdown, git, GitHub, R, and RStudio.

**2. R Programming Basics:**

Overview of R, R data types and objects, reading and writing data, Control structures, functions, scoping rules, dates and times, Loop functions, debugging tools, Simulation, code profiling

**3. 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|l.

**4. Exploratory Data Analysis:**

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

**5. 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.

### Reference Books:

- Rachel Schutt, Cathy O'Neil, "Doing Data Science: Straight Talk from the Frontline" by Schroff/O'Reilly, 2013.
- Foster Provost, Tom Fawcett, "Data Science for Business" What You Need to Know About Data Mining and Data-Analytic Thinking" by O'Reilly, 2013.
- John W. Foreman, "Data Smart: Using data Science to Transform Information into Insight" by John Wiley & Sons, 2013.
- Ian Ayres, "Super Crunchers: Why Thinking-by-Numbers Is the New Way to Be Smart" Ist Edition by Bantam, 2007.
- Eric Segel, "Predictive Analytics: The Power to Predict who Will Click, Buy, Lie, or Die", 1st Edition, by Wiley, 2013.
- 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 Science 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 22 22 22 22 23 23 23  
  - i. Find the median age of all students under 22 years
  - ii. Find the median age of all students
  - iii. Find the mean age of all students
  - iv. Find the modal age for all students
  - v. Two more students enter the class. The age of both students is 23. What is now mean, mode and median ?
11. Following table gives a frequency distribution of systolic blood pressure. Compute all the measures of dispersion.

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

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

14. Use Git / Github software to create Github account. Also, create a repo using Github.

## **DSE – 601 (c) Cloud Computing**

**Theory: 60 lectures**

### **1. Overview of Computing Paradigm (8 lectures)**

Recent trends in Computing : Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing,

### **2. Introduction to Cloud Computing (7 lectures)**

Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing,

### **3. Cloud Computing Architecture (20 lectures)**

Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service(IaaS), Platform as a Service(PaaS), Software as a Service(SaaS), How Cloud Computing Works, Deployment

Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.

### **4. Case Studies (13 lectures)**

Case study of Service model using Google App Engine, Microsoft Azure, Amazon EC2 , Eucalyptus.

### **5. Service Management in Cloud Computing (7 lectures)**

Service Level Agreements(SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.

### **6. Cloud Security (5 lectures)**

Infrastructure Security- Network level security, Host level security, Application level security, Data security and Storage- Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.

#### **Reference Books :**

- Cloud Computing Bible, Barrie Sosinsky, Wiley-India, 2010
- Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wile, 2011
- Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
- Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010
- Gautam Shroff, Enterprise Cloud Computing Technology Architecture Applications , Adobe Reader ebooks available from eBooks.com,2010
- Toby Velte, Anthony Velte, Robert Elsenpeter, Cloud Computing, A Practical Approach ,McGraw Hills, 2010.
- Dimitris N. Chorafas, Cloud Computing Strategies ,CRC Press, 2010

## **Cloud Computing Lab**

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms .
3. Working on tools used in cloud computing online-
  - a. Storage
  - b. Sharing of data
  - c. manage your calendar, to-do lists,

- d. a document editing tool
4. Exploring Google cloud
5. Exploring microsoft cloud
6. Exploring amazon cloud
7. Redhat cloud

## **DSE – 602 (a) Dissertation / Project work**

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.

## **DSE – 602 (b) Network Programming**

**Theory: 60 Lectures**

Transport Layer Protocols: TCP, UDP, SCTP protocol. Socket Programming: Socket Introduction; TCP Sockets; TCP Client/Server Example ; signal handling; I/O multiplexing using sockets; Socket Options; UDP Sockets; UDP client server example; Address lookup using sockets. Network Applications: Remote logging; Email; WWW and HTTP. LAN administration: Linux and TCP/IP networking: Network Management and Debugging.

### **Reference Books:**

- W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The sockets Networking API, Vol. 1, 3rd Edition, PHI.2003
- B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd.,2003
- Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education, 2nd Edition,2010
- R. Stevens, Unix Network Programming, PHI 2nd Edition,1990

## **Network Programming Lab**

**Practical: 60 Lectures**

Practical exercises based on concepts listed in theory.

## **DSE – 602 (c) Machine Learning**

**Theory: 60 Lectures**

### **1. Introduction:**

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

### **2. Softwares for Machine Learning and Linear Algebra Overview :**

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

### **3. Linear Regression:**

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

### **4. Logistic Regression:**

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

### **5. Regularization:**

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

### **6. Neural Networks:**

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

### **Reference Books:**

- Ethem Alpaydin, "Introduction to Machine Learning" 2nd Edition, The MIT Press, 2009.
- Tom M. Mitchell, "Machine Learning", First Edition by Tata McGraw-Hill Education, 2013.
- Christopher M. Bishop, "Pattern Recognition and Machine Learning" by Springer, 2007.
- 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 MATLAB/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/>).

Perform elementary mathematical operations in Octave/MATLAB like addition, multiplication, division and exponentiation.

Perform elementary logical operations in Octave/MATLAB (like OR, AND, Checking for Equality, NOT, XOR).

1. Create, initialize and display simple variables and simple strings and use simple formatting for variable.
2. 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.
3. 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.
4. Perform basic operations on matrices (like addition, subtraction, multiplication) and display specific rows or columns of the matrix.
5. 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.

6. 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.
7. Generate different subplots from a given plot and color plot data.
8. Use conditional statements and different type of loops based on simple example/s.
9. Perform vectorized implementation of simple matrix operation like finding the transpose of a matrix, adding, subtracting or multiplying two matrices.
10. 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.
11. 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.
12. 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.
13. Use some function for regularization of dataset based on problem 14.
14. 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.