

# SYLLABUS

## Curriculum Structures for UG syllabus for Information Technology (Honours)

**SUBJECT: INFORMATION TECHNOLOGY**



**Bodoland University**  
Debargaon, Kokrajhar (b.T.C.)

**Bodoland University**  
**Curriculum Structures for UG syllabus For Information Technology (Honours),**  
**No. of papers =14+12=26, Total Credits= 140**  
**Total Marks = 2400**

SL. NO.	CORE COURSE (14)	ABILITY ENHANCEMENT COURSE (AECC) (2)	SKILL ENHANCEMENT COURSE (SEC) (2)	ELECTIVE: DISCIPLINE SPECIFIC DSE (4)	ELECTIVE: GENERIC (GE) (4)
1	C1 :Computer Fundamental and Programming in C (4+4 lab)	(English/Hindi /MIL Communication)			GE 1: Information Communication Technologies Hardware
	C2: : Discrete Structures (5+1 lab)				
2	C3: Data Structure through C (4+4 lab)	Environmental Science			GE 2: Digital Logic
	C4 :Computer Oriented Numerical Methods (5+1 Lab/Tutorial)				
3	C5: Digital Logic & ICT (4+4 lab)		SEC 1: Core Java Programming		GE3: Introduction to Programming in C
	C6: Object Oriented Programming using C++				
	C7: database Management System (4+4 lab)				
4	C8: Data Communication & Networking (4+4 lab)		SEC 2: Digital Image Processing using MATLAB		GE4: Introduction to Database System.
	C9: Operating System (4+4 lab)				
	C10: Computer Organization and Architecture (5+1lab/ Tutorial)				

5	C11: Software Engineering (4+4 lab)			DSE 1: a)Microprocessor / b) Theory of Computation (5+1 lab/Tutorial)	
	C12: Web Technologies (4+4 lab)			DSE 2: a) Web Application Design using PHP Programming/ b)Data Mining & Warehousing	
6	C13: System Administration (4+4 lab)			DSE 3: (a)Compiler Design/ (b)System Analysis and Designing	
	C14: Computer Graphics (4+4 lab)			DSE 4: DISSERTATION/PROJECT WORK	

<b>SEM-I</b>						
<b>Paper Code</b>	<b>Course</b>	<b>L+T+P</b>	<b>Credit</b>	<b>End Sem</b>	<b>Int.</b>	<b>Total</b>
IT-101H	C-1	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20 20	100
IT-102H	C-2	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
Phy-GE-103H	GE-1	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20 20	100
COMM-104HR	<b>AECC-1:English/ Hindi/MIL (Communication)</b>	2	2	50(L)		50
<b>Total-</b>			<b>20</b>	<b>290</b>	<b>60</b>	<b>350</b>

<b>SEM-II</b>						
<b>Paper Code</b>	<b>Course</b>	<b>L+T+P</b>	<b>Credit</b>	<b>End Sem</b>	<b>Int</b>	<b>Total</b>
IT -201H	C-3	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT-202H	C-4	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
Phy-GE-203H	GE-2	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
ENV -204HR	<b>AECC-2: Environmental Science</b>	2	2	50(L)		50
<b>Total-</b>			<b>20</b>	<b>290</b>	<b>60</b>	<b>350</b>

SEM-III						
Paper Code	Course	L+T+P	Credit	End Sem	Int	Total
IT -301H	C-5	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -302H	C-6	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -303H	C-7	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -SEC1- 304HR	SEC-1	2	2	50(L)		50
Math -GE- 305H	GE-3	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
<b>Total-</b>			<b>26</b>	<b>370</b>	<b>80</b>	<b>450</b>

SEM-IV						
Paper Code	Course	L+T+P	Credit	End Sem	Int	Total
IT -401H	C-8	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -402H	C-9	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -403H	C-10	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -SEC2- 404HR	SEC-2	2	2	50(L)		50
Math-GE-405H	GE-4	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
<b>Total-</b>			<b>26</b>	<b>370</b>	<b>80</b>	<b>450</b>

<b>SEM-V</b>						
<b>Paper Code</b>	<b>Course</b>	<b>L+T+P</b>	<b>Credit</b>	<b>End Sem</b>	<b>Int</b>	<b>Total</b>
IT -501H	C-11	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -502H	C-12	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -DSE1- 503H	DSE-1	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -DSE2- 504H	DSE-2	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
<b>Total-</b>			<b>24</b>	<b>320</b>	<b>80</b>	<b>400</b>

<b>SEM-VI</b>						
<b>Paper Code</b>	<b>Course</b>	<b>L+T+P</b>	<b>Credit</b>	<b>End Sem</b>	<b>Int</b>	<b>Total</b>
IT -601H	C-13	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -602H	C-14	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -DSE3- 603H	DSE-3	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
IT -DSE4- 604H	DSE-4 (Project/Diss ertation)	4+0+2 5+1+0	6	60(L)+20(P) 60(L)+20(T)	20	100
<b>Total-</b>			<b>24</b>	<b>320</b>	<b>80</b>	<b>400</b>

1. Where there is a practical there will be no tutorial or vice-versa.
2. Institute should evolve a system/policy about ECA/ General Interest/Hobby/Sports/NCC/NSS/related courses on its own.
3. **Skill Enhancement Courses (SEC):** These courses are to be chosen from a pool of courses designed to provide value-based and/or skill-based knowledge and should contain both theory and lab/hands-on/training/field work. The main purpose of these courses is to provide students life-skills in hands-on mode so as to increase their employability. The list provided under this category are suggestive in nature and each Institution/College has complete freedom to suggest their own papers under this category based on their expertise, specialization, requirements, scope and need. However, in this case approval of Academic section of the University is mandatory.

## 1<sup>st</sup> SEMESTER

### **IT-101 C-1: COMPUTER FUNDAMENTALS AND PROGRAMMING IN C** **Theory: 60 Lectures**

#### **Unit 1: Fundamentals**

(15 Lectures)

Major components of a Digital Computer (A brief introduction of CPU, Main memory, Secondary memory devices and I/O devices) Keyboard, monitor, mouse, printers, Secondary storage devices (floppy disks, hard disks and optical disks), backup system and why it is needed ? Bootstrapping a Computer. Representation of numbers (only a brief introduction to be given) and characters in computer. ASCII. EDCDIC and Gray codes. Interpreter, Assembler, Linker and Loader. Definition and concepts of algorithm and its different implementations-pseudo code, flowchart and Computer programs.

**Number System:** Binary, Hexadecimal, Octal, BCD, and conversions of number systems. Representation of signed integers, Sign and magnitude, 1's complement and 2's complement representation. Arithmetic operations using 2's complement representation and conditions for overflow/underflow and its detection.

#### **Unit 2: Introduction to C**

(25 Lectures)

Elementary data types , variables, constants and identifiers. Integer, character floating point and string constants . variable declarations. Syntax and semantics. Reserved word. Initialization of variable during declarations Constant data types. Expression in C, precedence and associativity of C operators, unary, binary and ternary operators. C arithmetic operators, assignment operators, relational operators, logical operators and bit –wise operators . L-value and R-value. Side effects of operators. Expression statement. Conditional Statement-if, if-else, switch. Iterative Statement-white, do-while, for. Other

Statement –break , continue, goto, return, null Statement, block Statement. Function: function declaration. Calling a function. Parameters –Call by value, Call by reference and its absence in C. Recursion and how it works. Cast and sizeof operator. Automatic type Conversion.

Simple programs like programs to compute an arithmetic expression, unit conversion, the sum of a series ( like trigonometric series), GCD, factorial ( both recursive and non-recursive version ), fibonacci number (both recursive and non-recursive version), generation of prime numbers, reversing digits of an integer, finding the square root of a number, prime factors of an integer, base conversion of numbers, test if three points form a triangle and classify triangles as right angled, isosceles, equilateral etc., roots of a quadratic equation, generation of simple patterns of characters on screen.

### **Unit 3:Arrays and pointers**

(15 Lectures)

Storage classes : Automatic, External, Static, Register. Scope and lifetime of variables. Arrays and pointers and corresponding operators. Pointer arithmetic. Programs using arrays and pointers like sum, average, minimum, maximum of an array of numbers. Add and delete an element of an array. Merge two sorted arrays. String manipulation programs like string concatenation, palindrome, reverse, copy etc. Matrix manipulation like Sum of rows, columns, and diagonal elements of a matrix, transpose of a matrix.

### **Searching and Sorting**

Linear search, binary search. Selection sort and bubble sort.

### **Unit 4:Structures and Files**

(5 Lectures)

Structure – declaration and use. Structure member resolution and structure pointer member resolution operators. Programs to show the use of structure. Standard C library.

Files in C—opening, closing, reading and writing of files. Seeking forward and backward. Simple examples of file handling programs.

### **REFERNCE BOOKS**

1. Programming with C, B.S. Gottfried, Tata Mc-Graw Hill.
2. Programming in ANSI C, E.Balagurusamy, Tata McGraw – Hill
3. The C Programming Language, B.W. Kernighan and D.M.Ritchie, PHI
4. Computer Fundamentals, Anita Goel, Pearson, 2010.



**IT LAB-101 (C-I): COMPUTER FUNDAMENTALS AND PROGRAMMING IN C**  
**Lab Practical: 60 Lectures**

**At least 20 programming assignments have to be done by each student from the following list. The assignments should be selected in such a way that all the features of C language are included.**

1. Write a program to convert a given temperature value from Fahrenheit scale to Centigrade scale and vice versa.
2. Write a program to display ASCII value of a character.
3. Write a program to check whether a number is perfect or not.
4. Write a program to find out the biggest of three numbers using nested if.
5. A company insures its drivers if either of the following conditions are satisfied
  - Driver is married.
  - Driver is an unmarried, male and above 30 years of age.
  - Driver is unmarried, female and above 25 years of age.

Write a program to decide if a driver is to be insured using logical operators.

6. Write a program to read a list of positive integers terminated by -1 and display the odd and even numbers separately and also their respective counts.
7. Write a program to read values of n and x and print the value of y using switch case where
  - a.  $y=n+x$  when  $n=1$
  - b.  $y=1+x/n$  when  $n=2$
  - c.  $y= n+3x$  when  $n=3$
  - d.  $y=1+nx$  when  $n>3$  or  $n<1$ .
8. Write a program to n values of sales and then calculate the commission on sales amount where the commission is calculated as follows:
  - a. If sales  $\leq$  Rs.500, commission is 5%.
  - b. If sales  $> 500$  but  $\leq 2000$ , commission is Rs 35 plus 10% above Rs 500.
  - c. If sales  $> 2000$  but  $\leq 5000$ , commission is Rs 185 plus 12% above Rs.2000.
  - d. If sales  $> 5000$  ,commission is 12.5%.
9. Write a program to find out minimum, maximum, sum and average of n numbers without using array.
10. Program to find mean and standard deviation (SD) for a set of n numbers without using array.
11. Write a program to find out the roots of a quadratic equation. Use proper testing to find checks for real and complex roots.

12. Write a program to print the digits of a number in words. ( eg. if a number 841 is entered through the keyboard your program should print —Eight Four One.)
13. Write a program to print the PASCAL Triangle up to the n-th row where n is an input to the program.
14. Write a function to return the HCF of two positive integers. Write a main function to read two positive integers and print their HCF and LCM by using the above function.
15. Write a program to convert a decimal number into binary number using function.
16. Write a program to display the result of sine series using function.
17. Write a program to find the sum of the following series  
 $1+x-x^3/3!+x^5/5!-x^7/7!+\dots$  corrected up to the 3 decimal place.
18. Write a program to read n numbers in a sorted array and insert a given element in a particular position
19. Write functions to compute the factorial of a number using both recursive and non-recursive procedure.
20. Write a program to print the values of ncr and npr for given positive integers n r > 0. Use a function fact(n) to return the factorial of a non-negative integer.n.  
 $ncr=n!/r!(n-r)!$   $npr=n!/(n-r)!$
21. Write a program to display the first n Fibonacci numbers using function.
22. Write a program to display the prime numbers within a given range. Write a function to check whether a given integer is prime or not and use it.
23. Write a program to Multiply two matrices using function
22. Write a program to display the prime numbers within a given range. Write a function to check whether a given integer is prime or not and use it.
23. Write a program to Multiply two matrices using function
24. Write a program to display the upper Triangle and lower Triangle of a given square matrix using function.
25. Write a function to check if a given square matrix is symmetric or not. Write a main function to implement it.
26. Write a program to read a m X n matrix and calculate the Row sum and Column Sum of the matrix
27. Write a function to read in an integer and print the representation of the number using the sign and magnitude representation scheme using 8 bits. The program should check for overflow/under flow conditions. The left most bit is to be used as the sign bit.
28. Write a program to merge two sorted arrays.
29. Write a program to implement selection sort using function.
30. Write a program to count the number of vowels in a string.
31. Write a program to concatenate two strings using function (without using library function).
32. Write a program to convert a string from upper case to lower case and vice versa.
33. Write a program to swap two numbers using function (pass the pointers).
34. Write a program to sort n number of strings in ascending order using pointer.

35. Write a program using pointers to copy a string to another string variable (without using library function).
36. Declare a structure of a student with details like roll number, student name and total marks. Using this, declare an array with 50 elements. Write a program to read details of n students and print the list of students who have scored 75 marks and above.
37. Create a structure to store the following information of employees.  
 a. Employee's number, name, pay and date of joining.  
 It has been decided to increase the pay as per the following rules:  
 Pay  $\leq$  Rs.3000 : 20% increase  
 Pay  $\leq$  Rs.6000 but  $>$  Rs.3000 : 15% increase  
 Pay  $>$  Rs.6000 : no increase  
 Write a program to implement the above structure.
38. Write a program to read a text file and count the number of vowels in the text file.
39. Write a program to copy a text file to another file.

## **IT-102 (C-2): DISCRETE STRUCTURE**

### **Theory: 60 Lectures**

#### **Unit 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.

#### **Unit 2: Growth of Functions:** (8 Lectures)

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

#### **Unit 3: Recurrences** (10 Lectures)

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

#### **Unit 4: Graph Theory** (15 Lectures)

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

## 5. Propositional Logic

(12 Lectures)

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

### Recommended Books:

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

## IT LAB-102 (C 2): DISCRETE STRUCTURES

### Lab Practical: 15 Lectures

1. C programs to implement the Kruskal's algorithm to generate a minimum cost spanning tree.
2. C programs to implement the Prim's algorithm to generate a minimum cost spanning tree.
3. C program to implement Euler Circuit problem.
4. C program to implement Hamiltonian Cycle.
5. C Program to Evaluate Truth Value of Conjunction.
6. C program to Evaluate Truth Value of Disjunction.
7. Write a c program to print the union of two sets.
8. Write a c program to print the intersection of two sets.
9. Write a c program to print the Permutation of character string of two sets.
10. Write a c program to print the combination of a number of two sets.
11. Write a c program to find and print all the elements in the Cartesian product of two sets.
12. Write a c program to print the subtraction of two sets.
13. Write a c program to print the power of two sets.
14. Write a c program to print the proper subset of sets.
15. Write a c program to print the symmetric difference of two sets.

## **2nd SEMESTER**

### **IT-201 (C-3): DATA STRUCTURE THROUGH C**

**Theory: 60 Lectures**

**Unit 1:Arrays** (5 Lectures)

Single and Multi-dimensional Arrays, Sparse Matrices (Array and Linked Representation)

**Unit 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

**Unit 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

**Unit 4;Queues** (5 Lectures)

Array and Linked representation of Queue, De-queue, Priority Queues

**Unit 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)

**Unit 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).

**Unit 7:Searching and Sorting** (5 Lectures)

Linear Search, Binary Search, Comparison of Linear and Binary Search, Selection Sort, Insertion Sort, Insertion Sort, Shell Sort, Comparison of Sorting Techniques

**Unit 8:Hashing** (5 Lectures)

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

**Reference Books:**

1. Adam Drozdek, "Data Structures and algorithm in C++", Third Edition, Cengage Learning, 2012.
2. SartajSahni, Data Structures, "Algorithms and applications in C++", Second Edition, Universities Press, 2011.
3. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using C and C++.", Second edition, PHI, 2009.
4. Robert L. Kruse, "Data Structures and Program Design in C++", Pearson,1999.
5. D.S Malik, Data Structure using C++,Second edition, Cengage Learning, 2010.

6. Mark Allen Weiss, "Data Structures and Algorithms Analysis in Java", Pearson Education, 3rd edition, 2011
7. Aaron M. Tenenbaum, Moshe J. Augenstein, YedidyahLangsam, "Data Structures Using Java, 2003.
8. Robert Lafore, "Data Structures and Algorithms in Java, 2/E", Pearson/ Macmillan Computer Pub,2003
9. John Hubbard, "Data Structures with JAVA", McGraw Hill Education (India) Private Limited; 2 edition, 2009
10. Goodrich, M. and Tamassia, R. "Data Structures and Algorithms Analysis in Java", 4th Edition, Wiley,2013

### **IT LAB-201 (C-3): DATA STRUCTURES**

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

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration
12. (ii) WAP to display fibonacci series (i)using recursion, (ii) using iteration
13. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
14. WAP to create a Binary Search Tree and include following operations in tree:
  - (a) Insertion (Recursive and Iterative Implementation)
  - (b) Deletion by copying
  - (c) Deletion by Merging
  - (d) Search a no. in BST
  - (e) Display its preorder, postorder and inorder traversals Recursively
  - (f) Display its preorder, postorder and inorder traversals Iteratively
  - (g) Display its level-by-level traversals
  - (h) Count the non-leaf nodes and leaf nodes
  - (i) Display height of tree

- (j) Create a mirror image of tree
- (k) Check whether two BSTs are equal or not

15. WAP to convert the Sparse Matrix into non-zero form and vice-versa.
16. WAP to reverse the order of the elements in the stack using additional stack.
17. WAP to reverse the order of the elements in the stack using additional Queue.
18. WAP to implement Diagonal Matrix using one-dimensional array.
19. WAP to implement Lower Triangular Matrix using one-dimensional array.
20. WAP to implement Upper Triangular Matrix using one-dimensional array.
21. WAP to implement Symmetric Matrix using one-dimensional array.
22. WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
23. WAP to implement various operations on AVL Tree.

## **IT-202 (C-4): COMPUTER ORIENTED NUMERICAL METHODS**

### **Theory: 60 Lectures**

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

**Unit 2:** Bisection method, Secant method, Regula-Falsi method Newton-Raphson method, Newton's method for solving nonlinear systems Gauss elimination method (with row pivoting) and Gauss-Jordan method, Gauss Thomas method for tridiagonal systems

**Unit 3:** Iterative methods: Jacobi and Gauss-Seidel iterative methods Interpolation: Lagrange's form and Newton's form.

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

**Unit 5:** Numerical integration: Trapezoid rule, Simpson's rule (only method), Newton-Cotes open formulas.

**Unit 6:** Extrapolation methods: Romberg integration, Gaussian quadrature, Ordinary differential equation: Euler's method Modified Euler's methods: Heun method and Mid-point method, Runge-Kutta second methods: Heun method without iteration, Mid-point method and Ralston's method Classical 4<sup>th</sup> order Runge-Kutta method, Finite difference method for linear ODE.

#### **REFERENCE BOOKS:**

- [1] Laurence V. Fausett, Applied Numerical Analysis, Using MATLAB, Pearson, 2/e (2012)
- [2] M.K. Jain, S.R.K. Iyengar and R.K. Jain, Numerical Methods for Scientific and Engineering Computation, New Age International Publisher, 6/e (2012)
- [3] Steven C Chapra, Applied Numerical Methods with MATLAB for Engineers and Scientists,

Tata McGraw Hill, 2/e (2010).

**IT LAB-202 (C-4): COMPUTER ORIENTED 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.

**3<sup>rd</sup> SEMESTER**

**IT-301 (C-5): DIGITAL LOGIC AND INFORMATION COMMUNICATION TECHNOLOGY**

**THEORY: 60 LECTURES**

**Unit1: Boolean algebra and Logic gates**

(10 Lectures)

Boolean operators, axiomatic definition of Boolean algebra, Rules (postulates and basic theorems) of Boolean algebra, dual and complement of Boolean expression, Canonical form and Standard form, Sum of product and product of sum for m. Conversion between Boolean expression and truth table. Boolean expression and their simplification by algebraic method, Karnaugh map method (till four variable k-map) and Quine Mc Cluskey method, Don't care condition.

**Logic gates:**

Different types of gates, Implementation of logic expression with logic gates.

**Unit2: Combinational circuit**

(12 Lectures)

Adder: half adder, full adder, parallel binary adder, Subtractors: half subtracter and full subtracter, Magnitude comparator, Decoder, Encoder, Application examples of decoder and encoder, Multiplexer, Demultiplexer, Application examples of multiplexer and demultiplexer, programmable logic Array (PLA)

**Unit3: Sequential Circuit**

(10 Lectures)



Simple RS flipflop or latch, Clocked RS flipflop, D flipflop, JK flipflop, T flip flop, Edge triggered flipflop (SR, D, JK), Asynchronous preset and clear inputs, master Slave Flip Flop, JK Master slave flip flop., edge triggering and level triggering

#### **Unit 4: Evolution of computer system**

(5 Lectures)

Modern computer, Classification of computer, Personal Computer hardware: Monitor, Keyboard, Mouse, Scanner, printer, speaker.

#### **Unit 5 Hard Disk Drive**

(7 Lectures)

Logical structure and file system, FAT, NTFS. Hard disk tools: Disk cleanup, error checking, de fragmentation, scanning for virus, formatting, installing additional HDD. New trends in HDD. Floppy Disk Drive

#### **Unit 6: Optical Media**

(8 Lectures)

CDROM, theory of operation, drive speed, buffer, cache, CD-r, CD-RW, DVD ROM, DVD technology, preventive maintenance for DVD and CD drives, New Technologies. Driver installation, Writing and cleaning CD and DVD.

#### **Unit 7: Processor**

(8 Lectures)

Intel processor family. Latest trends in processor, Motherboard, Sockets and slots, power connectors. Peripheral connectors. Bus slots, USB, pin connectors. Different kinds of motherboards. RAM, different kinds of RAM. RAM up gradation. Cache and Virtual Memory concept.

#### **Recommended Books:**

1. Digital Logic and Computer Design, Morris M. Mano
2. Digital Fundamentals, Floyd and Jain
3. Comdex: Hardware and Networking Course Kit:: DreamTech press PC hardware: A beginners Guide: Ron Gilster: Tata Mc Graw Hill.

## **IT LAB-301 (C-5): DIGITAL LOGIC AND INFORMATION COMMUNICATION TECHNOLOGY**

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

1. Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.
2. Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva.
3. Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.
4. Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva.
5. Basic commands in Linux.
6. Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.
7. Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.
8. The test consists of various systems with Hardware / Software related troubles, Formatted disks without operating systems. Installation of antivirus software, configure their personal firewall and windows update on their computer. Then they need to customize their browsers to block pop ups, block active x downloads to avoid viruses and/or worms.

## **IT-302 (C-6): OBJECT ORIENTED PROGRAMMING USING C++**

### **THEORY: 60 LECTURES**

#### **Unit1: Principles of Object Oriented Programming**

(10 Lectures)

Basic concept of OOP, Procedural programming vs OOP. Introducing Object Oriented Approach relating to other paradigms. Benefits of OOP and methods. Applications of OOP. Abstraction, Encapsulation, Inheritance, Polymorphism, Review of C, Difference between C and C++.

**Unit2: Basic Elements and Ideas**

(10 Lectures )

Keywords and Identifiers in C++, Variables and Constants, Declaration and Initialization of Variables, Concept of Dynamic Initialization of variables, Enumerated variables, Basic data types, Arrays and Strings, User Defined Data types, Arithmetic, Relational, Logical Operators and Operator Precedence, Manipulators, Type Conversions and type cast operators, Console I/O: cin, cout functions. Control Statements.if; if-else; else....if; switch statements. Loops: for, while, do-while, Break, continue, go to. Pointer new, delete operators. Functions in C++: main function, function prototyping, call by reference, return by reference , functions inline, friend, virtual, library.

**Unit3: Classes and Objects**

(10 Lectures)

Encapsulation, information hiding, abstract data types, Object & classes, attributes, functions, C++ class declaration, member functions, State identity and behavior of an object, static data members and member functions, friend functions, constant member functions. Constructors and destructors, instantiation of objects, Default parameter value, object types.

**Unit4: Overloading**

(8 Lectures)

Function overloading: Function overloading with various data types, scoping rules for function overloading, Special features of function overloading. Operator overloading: Concept of Operator Overloading , overloading of Unary and Binary Operators., overloading binary operators using friends, manipulation of strings using operators.

**Unit5: Inheritance**

(8 Lectures)

Concept of inheritance, Types of Inheritance Single, multilevel, multiple, hierarchical, hybrid, virtual base class, abstract class. Type of Derivations – public, private & protected, Constructors in Derived Classes.

**Unit6: Virtual Functions and Polymorphism**

(5 Lectures)

Polymorphism, Categorization of polymorphism techniques: compile time polymorphism, Polymorphism by parameter, run time polymorphism pointers to derived class , Early binding vs. late binding, virtual function, pure virtual function.

**Unit7:Files and Exception Handling**

(5 Lectures)

Need For a Data File, Opening and Closing a File, Detecting Endof File, Classes for File Stream Operations , Sequential Input and Output Operations, Random Access File Processing. Concept and Use of Exception Handling. Throwing Mechanism, Catching Mechanism,Specifying Exception.

**Unit8: Templates**

(4 Lectures)

Function Template and Class Template.

**REFERENCE BOOKS**

- i) Herber t Schild, — The complete reference to C++||, Osborn McGraw Hill
- ii) R. Lafore, —Object Or iented Programming using C++||, Galgotia Publications
- iii) Ian Graham, —Object Oriented Methods||, Addison Wesley.
- iv) E. Balaguruswamy, "Object Oriented Programming with C++", Tata McGraw Hill 1997.
- v) D. ravichandran, —Programming With C++||, Tata McGraw Hill Publishing Company Ltd.

**IT LAB-302 (C-6): OBJECT ORIENTED PROGRAMMING USING C++**  
**Lab Practical: 60 LECTURES**

**Each student should do at least 12 assignments from the following list.**

1. Define a class named *triangle* to represent a triangle using the lengths of the three sides. Write a constructor to initialize objects of this class, given the lengths of the sides. Write member functions to check

(a) if a triangle is isosceles

(b) if a triangle is equilateral.

Write a main function to test your functions.

2. Define a structure **employee** with the following specifications.

**Empno**: Integer

**Ename**: 20 character

**Basic, hra, da**: float

**Calculate()** : a function to compute net pay as basic+hra+da with float return type.

**Getdata()**: a function to read values for empno, ename, basic, hra, da.

**Dispdata()**: a function to display all the data on the screen

Write a main program to test the program.

3. Define a class *circle* to represent circles. Add a data member *radius* to store the radius of a circle. Write member functions area() and perimeter() to compute the area and perimeter of a circle.

4. Define a class *complex* with two data members *real* and *imag* to represent real and imaginary part of a complex number.

Write member functions

**rpart()**: to return the real part of a complex number

**ipart()** : to return the imaginary part of a complex number

**Add()** : to add two complex numbers.

**Mul()** : to multiply two complex numbers.

Write constructors with zero, one and two arguments to initialize objects.

// This is an example of polymorphism.

5. Define a class *point* with two data members *xordinate* and *yordinate* to represent all points in the two dimensional plane by storing their x co-ordinate and y co-ordinate values.

Write member functions

**dist()**: to return the distance of the point from the origin.

**slope()**: to return the slope of the line obtained by joining this point with the origin.

Write constructors with zero, one and two arguments to initialize objects.

Write a friend function to compute the distance between two points.

6. Define a class *String* with the following data members

char \*p;

int size and write member functions to do the following (without using library function) and using dynamic memory allocation.

Length of the string

Compare two string

Copy one string to another.

Reverse the string.

Write suitable constructors and destructors. Write a copy constructor for the class.

7. For the class *complex* defined in 3 above, overload the <<, >>, + and \* operators in the usual sense. Also overload the unary – operator.

8. For the class *string* defined in 5 above, overload the <<, >> and + operators where + is to be used for concatenating two strings.

9. Define a class *time* to store time as hour, minute and second, all being integer values. Write member functions to display time in standard formats. Also overload the ++ and -- operators to increase and decrease a given time by one second where the minute and hour values will have to be updated whenever necessary.

10. Define a class to store matrices. Write suitable friend functions to add and multiply two matrices

11. Write a program to show the implementation of static members using class.

12. Define a class *student* with the following specification:

rollno :integer

sname :20 characters

Derive two classes *artsst* and *scst*. The class *artsst* will represent students belonging to Arts stream and the class *scst* will represent students belonging to science stream. The *artsst* class will have additional data members *ph, hs, en* and *as* to store marks obtained by a student in three subjects Philosophy, History, English and Assamese. The class *scst* will have additional data members *ph, ch, ma and eg* to store marks obtained in Physics, Chemistry, Mathematics and English.

Write the following member functions in the classes *artsst* and *scst*

*ctotal()* : A function to calculate the total marks obtained by a student

*takedata()* : function to accept values of the data members

*Showdata()* : function to display the mark sheet of a student .

13. Define an abstract base class *printer*. Derive three classes, *laser-printer*, *dmp*, *line-printer* and *ink-jet-printer*. The derived classes will have data members to store the features of that particular printer. Write pure virtual function *display()* in the base class and redefine it in the derived classes.

14. Define a abstract base class *figure* and add to it pure virtual functions  
*display()* to display a figure  
*get()* to input parameters of the figure  
*area()* to compute the area of a figure  
*perimeter()* to compute the perimeter of a figure.

Derive three classes *circle*, *rectangle* and *triangle* from it. A circle is to be represented by its radius, rectangle by its length and breadth and triangle by the lengths of its sides. Write a main function and write necessary statements to achieve run time polymorphism.

15. Write an interactive program to compute square root of a number. The input value must be tested for validity. If it is negative, the user defined function *my\_sqrt()* should raise an exception.

16. Define a class *rational* to store rational numbers as a pair of integers, representing the numerator and denominator. Write a member functions for setting the values of the numerator and denominator. This function should raise an exception if attempt is made to set a zero value as the denominator and in such cases it should be set to 1.

17. Write a class template for storing an array of elements. Overload the << and >> operators. Write a member function to sort the array in descending order.

18. Write a class template for representing a singly linked list. Write functions for inserting, deleting, searching and for displaying a linked list. Write a main function to test it on a linked list of integers and characters.

## IT-303 (C-7): DATABASE MANAGEMENT SYSTEM

Theory: 60 Lectures

### Unit 1:Introduction

( 6 Lectures)

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

### Unit 2:Entity Relationship(ER) Modeling

(8 Lectures)

Entity types, relationships, constraints.

**Unit 3:Relation data model**

(20 Lectures)

Relational model concepts, relational constraints, relational algebra, SQLqueries

**Unit 4:Database design**

(15 Lectures)

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

**Unit 5:Transaction Processing**

(3 Lectures)

ACID properties, concurrency control

**Unit 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.

**Books Recommended:**

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

**IT LAB-303 (C-7): 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 New
Location	Varchar(50)	Yes		Delhi

### Query List

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



employee was hired.

19. Query to display the following for each employee <E-Name> earns < Salary> monthly but wants < 3 \* Current Salary >. Label the Column as Dream Salary.
20. Query to display Name with the 1st letter capitalized and all other letter lower case and length of their name of all the employees whose name starts with J, 'A' and M.
21. Query to display Name, Hire Date and Day of the week on which the employee started.
22. Query to display Name, Department Name and Department No for all the employees.
23. Query to display Unique Listing of all Jobs that are in Department # 30.
24. Query to display Name, Dept Name of all employees who have an A in their name.
25. Query to display Name, Job, Department No. And Department Name for all the employees working at the Dallas location.
26. Query to display Name and Employee no. Along with their Manger's Name and the Manager's employee no; along with the Employees' Name who do not have a Manager.
27. Query to display Name, Dept No. And Salary of any employee whose department No. and salary matches both the department no. And the salary of any employee who earns a commission.
28. Query to display Name and Salaries represented by asterisks, where each asterisk (\*) signifies \$100.
29. Query to display the Highest, Lowest, Sum and Average Salaries of all the employees
30. Query to display the number of employees performing the same Job type functions.
31. Query to display the no. of managers without listing their names.
32. Query to display the Department Name, Location Name, No. of Employees and the average salary for all employees in that department.
33. Query to display Name and Hire Date for all employees in the same dept. as Blake.
34. Query to display the Employee No. And Name for all employees who earn more than the average salary.
35. Query to display Employee Number and Name for all employees who work in a department with any employee whose name contains a T.
36. Query to display the names and salaries of all employees who report to King.
37. Query to display the department no, name and job for all employees in the Sales department.

## **IT –304 (SEC-1): PROGRAMMING IN JAVA**

### **Theory: 60 Lectures**

#### **UNIT 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),

#### **UNIT 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.

### **UNIT 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.

### **UNIT4: 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.

### **UNIT 5: 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.

### **UNIT 6: Applets and Event Handling** (15 Lectures)

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

## **REFERENCE BOOKS**

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

10. David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.
11. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition, 2004.

## **IT LAB-304 (SEC-1): 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.

## **4<sup>th</sup> SEMESTER**

### **IT-401 (C-8): DATA COMMUNICATION AND COMPUTER NETWORKS**

**Theory: 60 Lectures**

#### **Unit 1: INTRODUCTION**

**(6 Lectures)**

Usage of Computer Network, study of topology, concept of protocol, Connection less and connection Oriented Service, Layered architecture, study of OSI and TCP model.

#### **Unit 2 : PHYSICAL LAYER**

**(10 Lectures)**

Introduction to Guided and Unguided media, physical description of twisted pair, coaxial cable, and fiber optic cable, Maximum data rate of a channel (Nyquist and shannons law), Basic concepts of Modulation and demodulation, Data encoding techniques (Manchester and Differential Mancestar encoding) . Network connecting devices hub, repeater, bridge, switch, router, and gateway

#### **Unit 3: DATA LINK LAYER**

**(7 Lectures)**

Functions and services of DLL, Framing and Framing Methods, Concept of Error Control, Error Correcting code(Hamming code), Error detecting code(CRC), Concept of Flow Control, Piggybacking, Stop-and-Wait sliding window protocol, Pipelining techniques(Go backN, Selective Repeat).

#### **Unit 4: MAC**

**(7 Lectures)**

What is MAC? Static Channel Allocation, Dynamic Channel Allocation, Pure ALOHA, Slotted ALOHA, Carrier Sense Protocol, 1-persistent CSMA, Non-PersistentCSMA, CSMA/CD, Ethernet(IEEE 802.3) and Ethernet Frame Format, Basic concept of Wireless LAN(IEEE 802.11), Binary Exponential Backoff Algorithm.

#### **Unit 5: NETWORK LAYER**

**(10 Lectures)**

Services and Functions of Network Layer, Virtual Circuit and Datagram Subnet, Routing, Distance Vector Routing, the Count-to-Infinity problem, Link State Routing, Congestion (definition and factors of congestion only), Definition of Quality of Service, Traffic shaping, Leaky Bucket and token Bucket Algorithm, Concept of IP Address.

#### **Unit 6: TRANSPORT LAYER**

**(10 Lectures)**

Functionality of transport Layer, Establishment and release of connection, TCP and UDP(Overview), Introduction to Sockets and socket primitives, port numbers.

#### **Unit 7: APPLICATION LAYER**

**(10 Lectures)**

Concept of E-mail, Telnet, WWW, DNS, HTTP, FTP, URL, SMTP, MIME.

### **REFERENCE BOOKS**

- 1.Red Hat Linux:Proffitt:PHI
- 2.Introduction to system Administration:IBM series:PHI

## **IT LAB-401 (C-8): DATA COMMUNICATION AND COMPUTER NETWORKS**

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

**Each student should do at least 4 assignments from the following list.**

Avoid using Loopback Communication for the assignments,1-4:

1. Write a server socket program using TCP/IP where the client side will send a request for an existing file to the server side and if the file exists in the server then send the contents of that particular file to the client in reply, otherwise display a message —file does not exist on the server, if exists displays the contents on the client side.
2. Develop a chat application using TCP/IP
3. Develop a client-server application using TCP/IP to input user's information and finally send them to the server and store there in a file.
4. Develop a server socket program where the client takes principal, rate of interest and number of years and send them to the server. In the server receive this information and find the simple interest and finally send the result to the client again and display it on the client's VDU.
5. Configure a Local Area Network (Wired/Ethernet) in Linux environment. Configure the network interface card using ifconfig command and also explore ping, ifdown and ifup commands.

## **IT-402 (C-9): OPERATING SYSTEM**

### **Theory: 60 Lectures**

#### **UNIT 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.

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

(6 Lectures)

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

#### **UNIT 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.

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

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

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

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

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

Policy mechanism, Authentication, Internal access Authorization.

**Recommended Books:**

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

**IT LAB-402 (C-9): OPERATING SYSTEMS**

**Lab Practical: 60 Lectures**

**C/ C++ programs**

1. WRITE A PROGRAM (using *fork()* and/or *exec()* commands) where parent and child execute:
  - a) same program, same code.
  - b) same program, different code.
  - c) before terminating, the parent waits for the child to finish its task.
2. WRITE A PROGRAM to report behaviour of Linux kernel including kernel version, CPU type and model. (CPU information)
3. WRITE A PROGRAM to report behaviour of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
4. WRITE A PROGRAM to print file details including owner access permissions, file access time, where file name is given as argument.

5. WRITE A PROGRAM to copy files using system calls.
6. Write program to implement FCFS scheduling algorithm.
7. Write program to implement Round Robin scheduling algorithm.
8. Write program to implement SJF scheduling algorithm.
9. Write program to implement non-preemptive priority based scheduling algorithm.
10. Write program to implement preemptive priority based scheduling algorithm.
11. Write program to implement SRJF scheduling algorithm.
12. Write program to calculate sum of n numbers using *thread* library.
13. Write a program to implement first-fit, best-fit and worst-fit allocation strategies.

**IT-403 (C-10): COMPUTER ORGANIZATION AND ARCHITECTURE**  
**Theory: 60 Lectures**

**Unit1: Introduction**

(10 Lectures)

Functional units of a computer, basic instructions, interconnection of functional units, bus structure, memory locations, memory addresses, memory operations, instruction and instruction sequencing (straight Line sequencing and branching), addressing modes, introduction to assembly language, stack, subroutine, I/O instructions.

**Unit2: Register Transfer Logic**

(8 Lectures)

Introduction, inter register transfer, arithmetic microoperation, logic microoperation, shift microoperation, Conditional control statements, fixed point binary data, instruction code, design of a simple computer.

**Unit3: Processor logic design**

(8 Lectures)

Processor organization, design of arithmetic and logic unit, status register, design of accumulator.

**Unit4: Control logic design**

(8 Lectures)

Hardware control, microprogrammed control block diagram, symbolic microprogram, microprogrammed CPU organization.

**Unit5: I/O Subsystem**

(13 Lectures)

Program controlled I/O, Interrupts: enabling and disabling interrupts, handling interrupts from multiple sources (priority control), DMA, structure and working of hard disk, CDROM, printer.

**Unit6: Memory subsystem**

(13 Lectures)

Semiconductor memory, SRAM, DRAM, ROM, speed size and cost, Cache memory, Mapping functions, replacement algorithms.

**Suggested Reading:**

1. Computer System Architecture, M.Morris Mano, PHI publication
2. Computer Architecture, Hamachar, Vranesic and Zaky.



## **IT-404(SEC 2): DIGITAL IMAGE PROCESSING USING MATLAB**

**Theory: 60 Lectures**

### **Unit 1. Introduction**

**(6 Lectures)**

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

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

**(7 Lectures)**

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

### **Unit 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.

### **Unit 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.

### **Unit 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



## **Unit 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.

## **Unit 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.

## **Unit 8. Image Segmentation**

**(9 Lectures)**

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

### **Reference Books**

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

## **IT LAB-404 (SEC-2) DIGITAL IMAGE PROCESSING**

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

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

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

## **5<sup>th</sup> SEMESTER**

### **IT-501 (C-11): SOFTWARE ENGINEERING**

**Theory: 60 Lectures**

#### **Unit 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).

#### **Unit 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.

#### **Unit 3: Software Project Management**

(8 Lectures)

Estimation in Project Planning Process, Project Scheduling.

#### **Unit 4: Risk Management**

(8 Lectures)

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

#### **Unit 5: Quality Management**

(8 Lectures)

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

#### **Unit 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.

#### **Unit 7: Testing Strategies & Tactics**

(8 Lectures)

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

#### **Recommended Books:**

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

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

**IT LAB (C-11): SOFTWARE ENGINEERING LAB**  
**Lab Practical: 60 Lectures**

S. No.	Practical Title
1.	Problem Statement, <ul style="list-style-type: none"> <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

**IT 502 (C-12): WEB TECHNOLOGIES**  
**Theory: 60 Lectures**

**Unit 1: Internet Basics:**

**3 Lectures**

History of the internet , the world wide web, getting connected ,web page, home page, web site, Internet services: email.

**Unit 2: Client Server Model:**

**15 Lectures**

Structure of an HTML document. HTML tags. The HTTP protocol details. Client side software. Web browsers (Netscape/Mozilla as example). DHTML Web server architecture and functions of an web server. JDBC and ODBC. Server side vs client side scripts, advantages and disadvantages of each. Client side and server side scripting languages and their uses. Dynamic web page. CGI scripts, Java Scripts and JSP as examples. PHP and Perl as scr ipting language. Browser plugins

**Unit 3: Web Object Model: CORBA**

**5 Lectures**

**Unit 4 XML:**

**12 Lectures**

Well formed XML syntax. references, well formed documents XML semantics. DTD, XML Schema, RELAXNG. Displaying XML on web. XML extensions. Processing XML files, Using a programming language and the SAX API,Using a programming language and the DOM API,Using a transformation engine and a filter ,Push Parsing ,Data binding ,Nonextractive XML

**Unit 5: Distributed Multitiered Application:**

**10 Lectures**

J2EE Components ,J2EE Clients, Web Components, Business Components. J2EE Containers, Container Services , Container Types, Enter prise JavaBeans Technology, Java Ser vlet Technology, JavaServer Pages Technology.

**Unit 6: Application server:**

**10 Lectures**

Persistence, Transaction processing, Concur rency control, Events using, Java Message Service, naming and directory services (JNDI),Secur ity ( Java Cryptography Extension (JCE) and JAAS ) Deployment of software components in an application server Remote procedure calls using RMIIOp.

**Unit 6 : Web Security:**

**5 Lectures**

Firewall, wrapper and Proxy.

**Suggested reading:**

1. The Internet Complete, M.L. Young
2. Using CGI by J. Dwight, M. Erwin, R. Niles.

3. Mastering JavaScript and Jscript by J. Jaworski
4. Dynamic HTML by D. Godmann.
5. Understanding HTML by D.P. Nagpal

## **IT LAB-502 (C-12): WEB TECHNOLOGY**

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

#### **HTML**

*(At least 12 assignments has to be done from this group)*

1. Create a HTML document consisting of HTML heading, paragraphs and images.
2. Create a HTML document and insert comments in the HTML source code and insert horizontal lines.
3. Construct HTML document to set the font of a text , size of the font, style of the font.
4. Create a HTML document to show how to create hyperlinks.
5. Create a HTML document to use an image as a link.
6. Create a HTML document to open link in a new browser window.
7. Create a HTML document to jump to another part of a document (on the same page).
8. Create a HTML document to insert images from another folder or another server.
9. Create an image-map, with clickable regions.
10. Create a HTML document with all table elements (Table, Caption, Table Row, Table Data element, Table Heading Element, THEAD, TFOOT, TBODY)
11. Create HTML document to make an unordered list, an ordered list, different types of ordered lists, different types of unordered lists, Nested list, Definition list.
12. Create HTML form with the all FORM elements (text fields, password field, Checkboxes, Radio buttons, Select elements, Drop-down list with a pre-selected value, Textarea (a multi-line text input field) and buttons.
13. Create HTML document with all Frame elements (FRAMESET, FRAME, NOFRAMES, and INLINE FRAME).
14. Create a HTML document to add AUDIO and VIDEO.
15. . Create a HTML document to aligning images (Let the image float to the left/right of a paragraph)
16. Create a HTML document to jump to a specified section within a frame

17. Construct a HTML document with CSS to Set the background colour of a page.
18. Construct a HTML document with CSS to set an image as the background of a page
19. Construct HTML document with CSS to Set the text color of different elements and align the text.
20. Construct HTML document to set different colours to visited/unvisited links, Specify a background colour for links

**XML Each student should do at least 2 assignments from the following list**

21. Construct an XML document that contain information about products of an organization.
22. Construct an XML document that contain information of 5 students (such as roll no., name , address, class).
23. Construct an XML document that contain details of 10 books.

**JAVAScript (At least 8 assignments has to be done from this group)**

24. Write a program in javascript to accept a name from the user and display the same name in an alert box.
25. Write a program in javascript to display a message in a confirm box.
26. Write a program in javascript to display the message ‘\_time is running out\_’ in the status bar.
27. Write a program in JavaScript to enter marks of a student and find his/her grade according to the following:  
  
if marks>=90 then grade A  
if marks>=80 then grade B  
if marks>=70 then grade C  
if marks>=60 then grade D  
else fail.
28. Write a program in JavaScript to create a button and when the button is clicked the message ‘\_Hello World\_’ is displayed on an alert box..
29. Write a program in JavaScript to accept 2 nos from the user and show the working of all arithmetic operators.
30. Write a program in JavaScript to accept 2 strings and concatenate them.
31. Write a program in JavaScript to display the current date and time.
32. Write a program in JavaScript to find the length of an array.
33. Write a program in JavaScript to check whether a string is palindrome or not.
34. Write a program in JavaScript that responds to a mouse click anywhere on the page  
( using mouse click).
35. Write a program in JavaScript to display the contents of a check box in a alert box.

36. Write a program to validate a form in the user id and password forms.
37. Write a program in JavaScript to create a welcome cookie, Button animation, Image map with added JavaScript Simple timing, Timing event in an infinite loop

**VBScript (*At least 4 assignments has to be done from this group*)**

38. Write a program in VBScript to create a variable.
39. Write a program in VBScript to uppercase to lowercase.
40. Write a program in VBScript to Create an array
41. Write a program in VBScript using conditional statements loop
42. Write a program in VBScript using loop.
43. Write a program in VBScript to display Date and Time
44. Write a program in VBScript to display the current month and day.

**ASP**

**Each student should do at least 2 assignments from the following list**

45. Write a program in ASP to interact with a user in a form that uses the "get" method.
46. Write a program in ASP to interact with a user in a form that uses the "post" method.
47. Write a program in ASP to interact with a user in a form with radio buttons.
48. Write a program in ASP to create a welcome cookie.

**IT-503 DSE-1(A): MICROPROCESSOR**  
**Theory: 60 Lectures**

**Unit1: Internal Organization of 8085A microprocessor** (10 Lectures)

User Programmable registers, PC, SP, accumulator, flags, data bus, address bus, control bus, instruction word size, opcode format, data format, memory addressing, I/O addressing, address decoding for memory and I/O.

**Unit2: 8085A microprocessor architecture** (15 Lectures)

Pinout of 8085A microprocessor, multiplexed address/data bus, control and status signal, demultiplexing of control signals, other signals, bus timings, fetch decode and execute cycle, timing diagram for opcode fetch memory read and memory write, interfacing memory and I/O.

**Unit3: Assembly language programming in 8085A microprocessor** (15 Lectures)

Complete instruction set in detail, programming examples, logic operation, counters and time delays, stack and subroutine, processing arrays, bit manipulation.



**Unit4: Interfacing**

(15 Lectures)

In and OUT instruction, decoding addresses, Interfacing LED, relay, seven segment display, switch, keyboard,.

**Unit5: Interrupts**

(5 Lectures)

Vectored interrupts, interrupt priorities, general purpose programmable peripheral devices, 8255A control and status registers, programming 8255A, introduction to 8279, 8254 and 8237 (block diagrams and basic functions).

**Suggested Reading :**

1. Microprocessor Architecture, Programming and Application with the 8085 by Ramesh S. Gaonkar.

**IT LAB-503 DSE-1(A): Microprocessor  
Lab Practical: 60 Lectures**

**Each student should do at least 8 assignments from the following list**

1. Write a program to add two 8 bit numbers & store it in a memory location 8820h.
2. Write a program to copy a block of memory from one location 8820h to another location 8840h.
3. Write a program to perform the addition of two 16 bit numbers.
4. Write a program to add two numbers & store it in a register e.
5. Write a program to load two unsigned numbers in register b & c. Subtract b from c. If the result is in 2's complement, convert the result in absolute magnitude & display it.
6. Write a program to find the difference of two numbers & store the result in 8830h.
7. Write a program to find the larger / smaller of two given numbers.
8. Write a program to subtract two numbers and add it to a given memory location.
9. Write a program to perform  $xy5$ , where x and y are 16bit numbers.
10. Write a program to find 2's complement of a number.

## **IT-503 DSE-1(B): THEORY OF COMPUTATION**

**Theory: 60 Lectures**

### **Unit1: Finite Automata**

(10 Lectures)

DFA, NFA, NFA with  $\epsilon$ -moves. Equivalence of DFA and NFA. Reduction of the number of states in a finite automata.

### **Unit2: Regular Languages and Regular Grammar**

(7 Lectures)

Concept of languages and grammar. Regular expressions. Connection between regular expressions and regular languages. Regular grammars, Right and LeftLinear Grammars. Equivalence between Regular languages and Regular grammars.

### **Unit3: Properties of Regular Languages**

(10 Lectures)

Closure under simple set operations union, intersection, concatenation, complementation and starclosure.proof of pumping lemma for regular language. Proof of nonregularity using Pigeonhole principle and using pumping lemma for regular languages.

### **Unit4: Context free languages**

(10 Lectures)

Contextfree grammars, leftmost and rightmost derivations, derivation trees. Parsing and Ambiguity in grammars and languages. Simplification of Context free Grammars removing useless productions, empty productions and unitproductions. Normal forms Chomsky and Greibach normal forms.

### **Unit5: Pushdown Automata**

(8 Lectures)

Definition and language accepted (acceptance by empty stack and final state and their equivalence).Pushdown Automata and Context free languages. Deterministic PDA and Deterministic Context free Languages.

### **Unit7: Properties of Context free Languages**

(5 Lectures)

Pumping Lemma for CFL. Using Pumping Lemma to show certain languages not to be Context free. Closure properties of CFL – closure under union, concatenation and star closure and showing that CFLs are not closed under intersection and complementation.

### **Unit8: Complexity Classes**

(10 Lectures)

Time and Space complexity of algorithms, average case and worst case analysis, asymptotic notation as a measure of algorithm complexity,  $O$ ,  $\Theta$ ,  $o$ ,  $\omega$ , and notations. Analysis of sorting algorithms Selection sort, Bubble sort, Insertion so t, Heap sort, Quick sort and analysis of searching algorithms – linear search and binary search.

### **Suggested Reading**

1. An introduction to Formal Languages and Automata, Peter Linz, Narosa.
2. Introduction to Automata Theory, Languages and Computation, Hopcroft and Ullman, Addison Wesley.

3. K. L. P. Mishra, N. Chandrasekaran; Theory of Computer Science (Automata, Languages and Computation), P. H. I.
4. T. H. Cormen, C. E. Leiserson and R. L. Rivest, Introduction to Algorithms, Tata Mcgraw Hill Publishers.

### **IT-504 DSE-2(A): WEB APPLICATION DESIGN USING PHP**

**Theory: 60 Lectures**

#### **UNIT I**

(8 Lectures)

Introduction to PHP as a programming Language: - Advantages of PHP, the server side architecture Decomposed, overview of PHP, history, object oriented support, benefits in running PHP as a server side script. Installing a web server, Internet information server, and IIS installation, testing web server setup.

#### **UNIT II**

(16 Lectures)

The basics of PHP: - data types, variables, constants, operators, Arrays, Conditional statements (if statement, Executing Multiple Statements, else if clause and switch statement), Iterations (for loop, while loop, controlling an array using a while loop, do while statement, foreach loop and special loop key words)

##### **String Manipulation and Regular Expression:**

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

#### **UNIT III**

(10 Lectures)

Functions, user defined functions, functions with arguments, built in functions( print(), includer(), header(), phpinfo() ), PHP server Variables, working with date and time , performing mathematical operations , working with string functions . System Variable (GET, POST, cookies& Session, Forums)

#### **UNIT IV**

(16 Lectures)

Working with forms, form elements (Text Box, Text Area, Password, Radio Button, Checkbox, The Combo Box, Hidden Field and image), adding elements to a form, uploading files to the Web Server using PHP, building a challenge and response subsystem and understanding the functionality of the FORM attribute Method Regular Expressions: - Engine, types of Regular Expressions, symbols used in Regular Expressions. Error handling in PHP: - Displaying errors, warnings, types of errors, error levels in PHP, logging Errors and Ignoring errors.

#### **UNIT V**

(10 Lectures)

Data base connectivity using PHP (MySQL, ODBC, ORACLE, SQL) Performing, executing Commands, different types of Data Base Operations like Insertion, deletion, update and query on data

#### **REFERENCE BOOKS**

1. Beginning PHP and MySQL,W. Jason Gilmore, Apress, 2010, Fourth Edition Froovice to Professional.
2. PHP6 and MySQL, Steve Suehring, Tim Converse and Joyce Park, Wiley India 2010, Second Edition.
3. HTML4 Complete ,E.Stephen Mask, Janan Platt BPB Publications, First Edition 1998.

## IT LAB-504 DSE-2(A): WEB APPLICATION DESIGN USING PHP

### Lab Practical: 60 Lectures

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

```
*  
* *  
* * *  
* * * *  
* * * * *
```
14. Write a simple PHP program to check that emails are valid.
15. WAP to print first n even numbers.
17. Using switch case and dropdown list display a —Hello message depending on the language selected in drop down list.

## **IT-504 DSE-2(B): DATA MINING AND WAREHOUSING**

**Theory: 60 Lectures**

### **Unit-I: Data Warehousing**

(15 Lectures)

Overview and concepts: Need for Data Warehousing, Basic elements of Data Warehousing, differences between Database Systems and Data Warehouse. Planning and Requirements: Project planning and management, collecting the requirements. Architecture and Infrastructure: Data Warehouse Architecture and its components, Infrastructure and metadata. Data Design and Data Representation: Principles of dimensional modeling, advanced topics data extraction, transformation and loading, data quality. Information Access and Delivery: Matching information to classes of users, OLAP in Data Warehouse, Data warehousing and the web. Implementation and Maintenance: Physical design process, Data Warehouse deployment, growth and maintenance.

### **Unit-2: Data Mining**

#### **Introduction**

(10 Lectures)

Basics of data mining, Different definitions of Data Mining and related concepts, Data mining process Data preparation, data cleaning and data visualization. KDD process. Data mining techniques: Clustering, Association rules and Decision trees.

### **Unit-3: Clustering**

(12 Lectures)

Partitional versus Hierarchical Clustering, types of data in clustering. Partitional clustering methods – kmeans, kmedoids, PAM, CLARA, CLARANS. Hierarchical clustering methods – BIRCH, CURE. Density based clustering methods DBSCAN. Categorical clustering – DBSCAN.

### **Unit-4: Rule Mining**

(12 Lectures)

What is an association rule? Mining association rules, frequent sets and border sets, algorithms for mining association rules – Apriori algorithm, PincerSearch algorithm, Border algorithm. Generalized association rule, quantitative association rule, association rule with item constraint.

### **Unit-5: Decision Trees**

(6 Lectures)

Introduction, tree construction principle, decision tree generation algorithms – CART, ID3.

### **Unit-6: Advanced Topics**

(5 Lectures)

(only basics of the following topics):

Web mining : Web Content Mining, Web Structure , Mining, Web Usage mining.Spatial mining, Temporal mining – Temporal association rules, sequence mining and GSP algorithm, discovery of frequent episodes.

## **REFERENCE BOOKS**

1. A.K. Puzari, Data Mining Techniques, University Press..
2. J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufman. 2001.

## 6<sup>th</sup> SEMESTER

### **IT-601 (C-13): SYSTEM ADMINISTRATION**

**Theory: 60 Lectures**

#### **Unit 1: Introduction**

(10 Lectures)

What is System Administration? Duties of a System Administrator. Basic features of the Linux operating system. Installation requirements, Partitioning the Hard drive in Linux, Installing the Linux system, Linux system Startup and Shutdown.

#### **Unit 2: Linux file system**

(14 Lectures)

Basics of Linux file system: hierarchy and types. commands *ls*, *cp*, *mv*, *rm*, *mkdir*, *rmdir*, *more*, *touch*, *ln*, *mount*, *umount*, *mkfs*., absolute and relative path names. Linux file types, attributes of file, setting user and group ownership of files and access permissions. mounting and unmounting file systems and partitions. Structure of /etc/fstab file and its purpose. hard link, symbolic link. Introductions of Shells. Linux environment variables. Basics of Shell Programming

#### **Unit 3: Linux Program and Processes**

(14 Lectures)

Program and Process. Foreground process, Background Process, daemons. Basic commands for starting and stopping processes. Examining the list of running processes on the system and understand the data presented there. cron, crontab file format, Standard I/O, Standard error ,redirection and piping.

#### **Unit 4: Managing user accounts**

(10 Lectures)

Adding a user, password, Creating Groups, adding and deleting groups, viewing user account information, structure of /etc/passwd, /etc/shadow, /etc/group files System monitoring and logging, Monitoring memory usage, disk space usage. Backup and Restore procedure

#### **Unit 5: IP and Server Configuration**

(12 Lectures)

IP address and IP address classes, subnetting, CIDR, Interface configuring with ifconfig, Gateway configuration, adding routes, ping, netstat, traceroute. Understanding the significance of the /etc/services file and well known port numbers. Server configuration DHCP, NFS, NIS, SAMBA, PROXY..

#### **REFERENCE BOOKS**

1. Red Hat Linux:Proffitt:PHI.
2. Introduction to system Administration:IBM series:PHI.
3. Essential System Administration:Frisch:O'REILLY

**IT LAB-601 (C-13) SYSTEM ADMINISTRATION**  
**Lab Practical: 60 Lectures**

Each student should do at least 15 assignments from the following list.

1. Installation of Linux Operating System and partitioning the disk.
2. Installing software packages in linux OS using GUI as well as command line.
3. Changing the default run level of a system
4. Mounting and unmounting a removable media.
5. Finding the list of all running processes and redirect the output in a file.
6. Use of different kill signals to kill a running process.
7. Bringing a process from back ground to fore ground and vice-versa.
8. Adding and managing user accounts.
9. Monitoring disk space quota and memory usage and redirect the output in a file.
10. Backup and restoring a file.
11. Compression and extracting a file. Use command line.
12. Configuring a network interface and assigning a default route.
13. Scheduling job using crontab.
14. Changing the ownership and access permission of file or directory. Use command line.
15. Copy, move and rename a file.
16. Configuring a ftp server
17. Assigning address of DNS.
18. Use of ssh, telnet, netstat, ping, route commands.
19. Use grep, awk, sed commands.
20. Use of redirection and piping.
21. Monitoring and managing system log information.
22. Basics of firewall using iptables.
23. Basics of configuring http server.
24. Managing different services in linux.
25. Monitoring the traffic going through a network interface.
26. Write shell script to
  - a. Find factorial of a given number
  - b. Convert a decimal number to hexadecimal number

**IT-602 (C-14): 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**

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



**IT LAB-602 (C-14) COMPUTR 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.

**IT-603 DSE-3(A) COMPILER DESIGN**  
**Theory: 60 lectures**

**Unit 1: Introduction** (10 Lectures)

What is a compiler? Phases of compiler. Overview of working of a compiler, linker, loader.

**Unit 2: Lexical Analysis** (13 Lectures)

NFA, DFA, conversion from NFA to DFA. Regular expression. Regular expression to NFA conversion. Minimization of DFA. ,Structuer of Lexical analyzer ,use of finite autometa to write lexical analyser .

**Unit 3: Syntax analysis** (12 Lectures)

Grammar representation. Derivation and parse tree. Ambiguity and possible elimination. Top down parsing. Recursive descent and predictive top down parsing. Elimination of Left recursion. Bottom up parsing. Operator precedence parsing, LR parsing (including SLR and LALR). Er ror detection and recovery. Parser table construction.

**Unit 4: Code generation** (15 Lectures)

Symbol table contents, implementation. Type checking. Syntax directed translation. Forms of intermediate codes. Abstract Syntax Trees, Directed Acyclic Graph, Three address code. Intermediate code generation for different language constructs , boolean expressions, if, ifelse, while, case or switch. Target code generation issues, registerallocation, Runtime storage management

**Unit 5: Code Optimisation** (10 Lectures)

DAG, basic blocks, Common subexpression elimination, variable propogation, code motion, strength reduction, elimination of dead code, loop optimisation.

## REFERENCE BOOKS

1. Aho, Sethi, Ullman; Compilers, Principles, Techniques, Tools, Pearson Education.
2. Compiler Design, Santanu Chattopadhyay, P.H.I.

## IT-603 DSE 3(B): SYSTEM ANALYSIS AND DESIGNING Theory: 60 Lectures

### Unit 1: Introduction

(5 Lectures)

System Definition and Concepts: General Theory systems, Manual and automated Systems, Reallife Business Sub –Systems. System .Environments and Boundaries .Real Time and distributed systems. Basic principles of successful systems. Approach to system Development: Structured system Analysis and Design, Prototype, joint Application Development.

### Unit 2: Systems Analyst

(5 Lectures)

Role and Need of Systems Analyst. Qualification and responsibilities .System Analysis as a profession

### Unit 3: System Development Cycle

(8 Lectures)

Introduction to Systems Development Life Cycle(SDLC) Various phases of SDLC: Study, Analysis, Design, Development, Implementation, Maintenance Systems documentation consideration: principles of systems Documentation, Types of documentation and their importance, Enforcing documentation discipline in an organization.

### Unit 4: System Planning

(8 Lectures)

Data and fact gathering techniques: interviews, Group Communication Questionnaires, presentations & Site visits. Assessing project Feasibility: Technical, Operational, Economic, Cost Benefits Analysis, Schedule Legal and contractual, political. Modern methods for determining system requirements: joint Application, Development program, prototyping, Business process Reengineering. System Selection plan and proposal.

### Unit 5: Modular and Structured Design

(5 Lectures)

Module Specifications. Top –Down and bottom- up design. Module coupling and Cohesion. Structure charts.

### Unit 4: System design and Modeling

(10 Lectures)

Process Modeling, logical and physical design, Conceptual Data Modeling : entity Relationship Analysis, Entity –relationship Modeling, ERDs and DFDS Concepts of Normalization. Process Description: Structured English, Decision Thee, Decision Tables Documentation: Data Dictionary Recording Data Description

### Unit 6: Input and output

(7 Lectures)

Classification of forms, input/ output forms design. User interface design, Graphical interfaces. Standards and guidelines for GUI Design. Designing physical Files and databases: Designing Fields, Designing Physical Records , Designing Physical Files, Designing Databases. Introduction to CASE Tools, Features Advantages and Limitations of CASE Tools, Awareness about some commercial CASE Tools.

**Unit 7: System Implementation and Maintenance** (7 Lectures)

Planning Consideration. Conversion methods, procedures and controls. System acceptance criteria. System Evaluation and performance. Testing and Validation. Preparing User Manual. Maintenance Activities and Issues.

**Unit 8: Computer System Audit and Security** (5 Lectures)

Audit of Computer System Usage. Types of Threats to Computer System and Control Measures: Threat and Risk Analysis, Disaster Recovery and Contingency planning, viruses.

**REFERENCE BOOKS**

1. J Hoffer,||Modern Systems Analysis and Design||, Joey George and Joseph Valacich, Pearson Education.
2. A Dennis and BH Wixom,||System Analysis and Design|| John Wiley and Sons, Inc.

**IT-604 DSE-4: Dissertation / Project work**

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

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

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

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

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

-----