DEVI AHILYA VISHWAVIDYALAYA, INDORE

Detailed Syllabus of

M.Sc.(CS)

Effective From July – 2011

TOTAL PAGES = 52

Effective for Students Admitted in July, 2011 onwards

M.Sc. Computer Science (CS)
Scheme of Examination
(Effective from July 2011)

I Semester:

Course Code	Title	Scheme of marks		Total
		Theory	Practical	
CS-101	Computer and Communication Fundamentals	50	-	50
CS-102	Programming and Problem Solving Using C	50	50	100
CS-103	Operating Systems	50	50	100
CS-104	Discrete Structures	50	-	50
IC-105	Communication Skills	50	-	50
Total		250	100	350

(Effective from Jan 2012)

II Semester:

Course Code	Title	Scheme of marks		Total
		Theory	Practical	
CS-201	Data Structures using C++	50	50	100
CS-202	Database Management System	50	50	100
CS-203	Software Engineering	50	-	50
CS-204	Computer Architecture	50	-	50
CS-205	Computer oriented numerical and statistical method	50	-	50
Total		250	100	350

(Effective from July 2012)

III Semester:

Course Code	Title	Scheme of marks		Total
		Theory	Practical	
CS-301	Object Oriented Programming using JAVA	50	50	100
CS-302	Database Application and Tools	50	50	100
CS-303	Theory of Computation	50	-	50
CS-304	Computer Graphics and Multimedia	50	50	100
CS-305	Computer Networks	50	-	50
Total		250	150	400

(Effective from Jan 2012)

IV Semester:

Course Code	Title	Scheme of marks		Total
		Theory	Practical	
CS-401	Linux/Unix Administration	50	50	100
CS-402	Compiler Design	50	-	50
CS-403	Internet & Web Technology	50	50	100
CS-404	Design and Analysis of Algorithms	50	50	100
CS-405	Major Project		50	100
Total		200	200	400

Scheme for award of degree shall be same as for MBA (CM)

Minimum passing mark in theory shall be 40%. Aggregate pass percentage shall be 45 %. The result shall be declared as PASS/ FAIL (No supplementary or ATKT exam). If the aggregate percentage is >=45 % and even if student is failing in one or more subject, then the result shall be declared as PASS.

On the basis of aggregate of the total marks obtained in all semesters, the division will be decided as under:

1) First Division 60 % or above
2) Second Division 50 % or above

On the basis of Aggregate of total marks obtained in two years.

CS- 101 Computer and Communication Fundamentals

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	Computer Organization: Digital and Analog computers, Major components of a digital computer, Memory addressing capability of a CPU, Word length of a computer, Processing speed of a CPU, Definitions of Hardware, Software and Firmware. Definitions of Dumb, Smart and Intelligent terminals. Binary Systems: Digital Systems, Binary Numbers, Number Base Conversions, Octal and Hexadecimal Numbers, Complements, Signed Binary Numbers, Binary Codes: BCD code, Gray Code, ASCII code, Excess 3 Code, Error detecting Code. Computer Arithmetic: Binary representation of Negative Integers using 2's complement and Signed magnitude representation, Fixed point Arithmetic operations on Positive and Signed (Negative) Integers like addition, subtraction, multiplication, Booth algorithm for multiplication, and bit pair (fast) multiplication. Division of positive and negative binary numbers.
Unit 2	Boolean Algebra and Logic Gates: Basic Definitions, Basic Theorems and properties of Boolean Algebra, Boolean Functions, Canonical and standard forms, Other Logic operations, Digital Logic gates, Integrated Circuits. Gate-Level Minimization: The K-Map Method, 3 and 4 variable K-Map, Product of sums simplification, Sum of Products simplification, Don't care conditions, NAND and NOR implementations, Exclusive-OR function.
Unit 3	Combinational Logic: Combinational Circuits, Analysis Procedure, Design Procedure, Binary half adder, binary full adder, binary full subtractor, binary parallel adder, carry propagation delay and Propagation delay calculation of various digital circuits. Fast adder, Decimal Adder, Binary multiplier, magnitude comparator, Parity generator, seven segment display, BCD to excess three code converter, Decoders, Encoders, Multiplexes, and Demultiplexers Synchronous Sequential logic: Sequential circuits, Latches, Flip Flops: SR, D, JK, T. Master Slave JK Flip flop. Characteristic equations and Excitation tables of flip-flops. Analysis of clocked sequential circuits: State diagrams, State equations for D, JK and T Flip flops. State reduction methods using all Flip Flops. Mealy and Moore Models.
Unit 4	Shift Registers- Serial in Serial out, Serial in Parallel out, Parallel in Serial out and Parallel in Parallel out. Designing of Asynchronous (Ripple) Counters, Design of Synchronous Counters. Signal characteristics: frequency, phase and amplitude, Types of communication with and among computers, Characteristics of communication channels, allocation of channel, signal power, amplitude and frequency modulation, space division multiplexing, frequency division multiplexing, Time division multiplexing.
Unit 5	Physical communication media: bounded media and unbounded media, Infrared technology, public switched telephone networks, digital modulation: ASK, PSK and FSK, half duplex and full duplex transmission, asynchronous and synchronous transmission. Need for computer communication networks, Internet and world wide web: E-mail, File transfer, Remote login. Networking topologies. LAN, MAN, WAN, Ethernet LAN. Layer 2 and layer 3 switches, Wireless LAN. Switch hub and router. Wireless LAN, interconnecting networks, IP address and IP data grams, Internet and intranet; inter security, virtual private networks, future of Internet technology.

1. Text Books :
   1. Digital Design by M. Morris Mano.
      Publication: PHI Eastern economy edition (Year-2001)
   2. Computer Architecture By Nicholas Carter, Schaum Series Adaptation, end edition, 2011 .
2. Reference books :
   1. Data communications and networking By A. Forouzan
      Publication: TMH Third edition (Year- 2004)
   2. Computer Fundamentals – Architecture and Organization By B. Ram.
   3. Computer networks by Andrew Tanenbaum
      Publication: PHI Fourth edition (Year- 2003)
   4. Computer organization and architecture by William Stallings.

Effective for Students Admitted in July, 2011 onwards

Class Assignments:

1. Write the various units of CPU. Describe the working of each unit.
2. Write the definition of hardware, software and firmware.
3. What you mean by word length of a computer.
4. If address bus for a CPU is 20 bits wide, then calculate the size of memory that can be attached to it.
5. If data bus for a CPU is 16 bit then what will be its word length.
6. What are the major factors that decide the processing speed of a CPU?
7. Write the first 20 decimal digits in base 4.
8. Write the first 20 decimal digits in base 3.
9. Add and multiply the following numbers in the given base without converting to decimal.
   I. $(1230)_4$ and $(33)_4$
   II. $(130)_5$ and $(34)_5$
   III. $(230)_6$ and $(54)_6$
   IV. $(130.4)_5$ and $(34.4)_5$
10. Write the first 100 decimal digits into binary.
11. Convert the following numbers into binary.
    I. 123.56 II. 456.75
    III. 345.9 IV. 890.9
    V. 567.9 VI. 668.7
12. Converts the following numbers into decimal.
    I. 10101010 II. 101010110011
    III. 10110101.1111 IV. 101010111.1101
13. Perform the following conversion , without converting into decimal :
    I. $(3674)_8$ to $( )_{16}$ to $( )_2$
    II. $(1001010101010)_2$ to $( )_{16}$ to $( )_8$
    III. $(AC4)_{16}$ to $( )_2$ to $( )_8$
14. Represent the following numbers into sign magnitude representation.
    I. -11 II. 15 III. –15 IV. 7 V. –32
15. Represent the following numbers into sign two's complement representation.
    I. -11 II. 15 III. –15 IV. –32
16. Perform the M-N and N-M using two's complement method.
    I. M= 101100101 and N = 111000110
    II. M =101 and N = 110110
    III. M= 45 and N =90
17. Perform the multiplication of following numbers using two's complement method
    I. 16 *-6
    II. 23*-9
    III. –4 * -6
18. Perform the multiplication of following numbers using booth algorithm.
    I. 12*5 II. 30*-5 III. -13*25
19. Simplify the following Boolean functions to minimum number of literals
    I. AB+AB’+C’+ABC
    II. ABC+ABC’+AB
    III. AC+BC+ABC+BC’
    IV. A'CD + A'C'D' + A'B'C'D +ABC' +ABCD +A'B'C'D'
20. Simplify the following functions using karnaugh map.
    I. F=□(1,4, 7, 8)
    II. F=□(3,4, 7)

III. F=□(0,1,2,4, 7, 8,10,15,)
IV. F=□(1,4, 7, 8,10) and D=□(2,11,12)
V. F=□(1,2,3,4,7,8,9,11,12,15)

21. Implement the following function using AND and OR gate.
    I. F=□(1,4, 7, 8)
    II. F=□(3,4, 7)
    III. F=□(0,1,2,4, 7, 8,10,15,)
    IV. F=□(1,2,3,4,7,8,9,11,12,15)
22. Implement the following function using only NOR gate.
    I. AB+AB'+C'+ABC
    II. ABC+ABC'+AB
    III. AC+BC+ABC+BC'
    IV. ABC'+BC+AB
23. Implement the following function using only NOR gate.
    I. AB+AB'+C'+ABC
    II. ABC+ABC'+AB
    III. AC+BC+ABC+BC'
    IV. ABC'+BC+AB
24. Design a combinational circuit that accepts a three-bit number and generates an output binary number equal to the square of the input number.
25. Design a combinational circuit that accept BCD values and generate cube of that number.
26. Design a combinational circuit that generates 540321 weighted code for a BCD input.
27. Design a combinational circuit with four input lines that represent a decimal digit in BCD and four output lines that generate the 9's complement of the input number.
28. Design a 4-bit circuit that generates booth multiplier for given multiplier.
29. A combinational circuit is defined by the following two functions.
    I. F_1 = x'y'+xyz'
    II. F_2 = x'+y
    III. F_3 = xy+x'y'
30. Design the circuit with a decoder and external gates.
31. Design an even parity generator.
32. Design 3 to 8 decoder by using two 2 to 4 decoder.
33. Design 4 to 16 decoder by using two 3 to 8 decoder.
34. Design 5 to 32 decoder by using four 3 to 8 decoder.
35. Derive the excitation table of all flip-flops with their truth table.
36. Design a synchronous down counter.

CS - 102 Programming and Problem Solving Using C

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	Introduction to Computer based Problem Solving, Classification of programming languages, Programming Environment {Assemblers, compilers, interpreters, linkers, and loaders}. Introduction to Computer based Problem Solving, Features of good programs – Structured program, modular program, fundamentals of programming. Classification of programming languages: low-level language, middle-level language, high -level language along with examples and differences. Programming Environment: Assemblers, compilers, interpreters, linkers, and loaders. Programming Concepts with Flowcharts and algorithms, How to make flowchart: Introduction of the symbols of flowchart and algorithm. Importance of Flowchart and algorithms. Solving and discussing problems with flowchart and algorithm. Class assignment on flowchart. Developing and debugging flowcharts for Programming Problem.
Unit 2	Overview of C: C character set, identifiers and keywords. Data types (description of different types of data types along with examples.), Coding style (first program of C). Format Specifiers associated with each data type. Constants and Variables: Some examples of constants and variables. Expressions and Operators: Instructions, declaration, arithmetic, unary, relational, logical, assignment and conditional operators. Basic Input/ Output functions like scanf() and printf() along with examples. Introduction of Formatting Characters and how to use them along with examples. Sequence Control Structures, Decision Control Structures (ifelse and nested if-else) along with examples. Some exercises based on decision control statements. Loop Control Structures: Introduction of loops like for loop and some examples of “for loop”. Examples of do-while loop, while loop, break, continue. Explanation and examples of Case Control Structures. Nesting of loops i.e. loops in a loop along with examples. Introduction of Arrays: Array initialization, bound checking. Programs using 1D Array. Advantages and disadvantages of Arrays. 2D and multidimensional array and programs based on it.
Unit 3	Introduction of Pointers: Pointers declaration, pointer arithmetic and operation. Some programs based on pointers. Concept of Functions: prototype and parameter passing and some programs based on it. Advantages/pitfalls of pointers. Dynamic Memory allocation: malloc() and calloc() functions. Pointers and their Applications: String Handling, Matrices Operations. Call by reference and Call by value. String Handling: Pointers and strings, standard library string functions, pointer to array of characters vs. string and its limitation. Matrices Operations along with examples. Introduction of Storage classes (register, static, auto and extern) and their differences.Introduction of Structures: Structure declaration, program and application.Introduction of Union: Union declaration, program and application.Difference between Structure and Union.Some standard Functions like gets(), goto(), puts(), getchar(), putchar(), getche() etc.
Unit 4	Introduction of user defined functions along with examples. Functions with arguments, without arguments along with the examples. Communication between functions (how one function calls the other functions). Some exercise based on functions. Arguments to main: Introduction of argc and argv. Enumerations and bit fields and program based on it. Introduction of bit wise operators along with examples. Introduction of Recursion and program based on it. Concept of typedef and its programming.
Unit 5	Concept of Pre-Processors: macro expansion, file inclusion Conditional compilation. Macros with argument and macro versus function. Concept of local and global variables. Header file creation. Introduction to file handling: Different operations on file like read, write and append. File Creation and programs based on file creation, reading and merging. Advanced pointer. Other Programming Techniques.

1. Text Book :

   Let us C, Yashwant Kanetkar, BPB Publications.

Effective for Students Admitted in July, 2011 onwards

2. Reference books :
    1. B.W. Kernighan & D.M. Ritchie, "The C Programming Language", Prentice Hall of India.
    2. Ashok N. Kamthane , “Programming with ANSI and Turbo C”, Pearson Education
    3. Ashok N. Kamthane et. al., Computer Programming and IT (for RTU), Pearson Education, 2011
    4.

Class Assignments:

1. Write a C program to display "Hello Computer" on the screen.
2. Write a C program to display Your Name, Address and City in different lines.
3. Write a C program to find the area of a circle using the formula: Area = PI * r2
4. Write a C program to find the area and volume of sphere. Formulas are: Area = 4*PI*R*R Volume = 4/3*PI*R*R*R.
5. Write a C program to print the multiply value of two accepted numbers.
6. Write a C program to read in a three digit number produce following output (assuming that the input is 347)
   3 hundreds
   4 tens
   7 units
7. Write a C program to read in two integer values and display one as a percentage of the other. Typically your output should look like 20 is 50.00% of 40 assuming that the input numbers were 20 and 40. Display the percentage correct to 2 decimal places.
8. Write a C program to find out whether the character presses through the keyboard is a digit or not (using conditional operator).
9. Write a C program to swap variable values of i and j.
10. Write a C program to find the maximum from given three nos.
11. Write a C program to find that the accepted no. is Negative, Positive or Zero.
12. Write a program which reads two integer values. If the first is lesser print the message up. If the second is lesser, print the message *down* if they are equal, print the message *equal* if there is an error reading the data, print a message containing the word *Error*.
13. Write a C program that prints the given three integers in ascending order using if –else.
14. Given as input three integers representing a date as day, month, year, print the number day, month and year for the next day's date. Typical input: “28 2 1992” Typical output: “Date following 28:02:1992 is 29:02:1992”.
15. Write a C program for calculator designing using switch /case loop.
16. Write a C program to convert decimal to binary.
17. Write a C program to convert decimal to hexadecimal.
18. Write a C program to display first 25 Fibonacci nos.
19. Write a C program to display first 100 prime nos.
20. Write a C program to find factorial of accepted no.
21. Write a C program to find the sum of digits of accepted no.
22. Write a C program to print the accepted no and its reverse no.
23. Write a C program to print all the factors of accepted no.
24. Write a C program to find HCF of two given numbers.
25. Write a C program to find LCM of two given numbers.
26. Write a C program to find all the prime numbers between two given numbers.
27. Write C programs to print the terms of each of the following series:
    i. Sin(x) ii. Cos(x) iii. Log (1+x) iv. log (1-x) v. e_x vi. e_{-x}

28. Write a C program to print the sum of series.(will be given in class)

29. Display the following output on screen (assuming the value for input parameter n=5) :

a. * ** *** **** *****	b. 1 12 123 1234 12345	c. AA B ABC ABCD ABCDE	d. 1 23 345 4567 56789	e. 1 23 456 78910 101112131415
f. ***** **** *** ** *	g. ABCDE ABCD ABC AB A	h. * *** ***** ******* *********	i. 1 123 12345 1234567 123456789	j. 1 121 12321 1234321 123454321
k. 1 01 101 0101 10101	l. 1 22 333 4444 55555	m. A AB ABC AB A	n. ABCDEDCBA ABCD DCBA ABC CBA AB BA A A	o. 1 121 12321 1234321 123454321

30. Write a C program to find minimum, maximum, sum and average of the given one dimensional array.

31. Write a C program to perform the basic Matrix operations addition, subtraction, multiplication, Transpose.

32. Write a program to take a sentence as input and reverse every word of the sentence.

33. Write a C Function for the following task
    a) Calculating Factorial
    b) Nth Fibonacci number
    c) Swapping of two variables
    d) Minimum/maximum value from the given input values.

34. Write User Defined Function and test them in the main program for the following standard function
    a) int mystrlen(char *s)
    b) char *mysubstr(char *s, int i, int j)
    c) char *mystrcat(char *s1,char *s2)
    d) int mystrcmp(char *s1, char *s2)
    e) char *mystrrev(char *s)
    f) int mystrend(char *s, char *t)
    g) char *myreplace(char *s, char *old, char *new)
    h) char *mytoupper(char *)
    i) char *mytolower(char*)

35. Write the following recursive C Function
    a) Factorial of a given number
    b) Nth Fibonacci number
    c) Reverse of a given String
    d) Reverse of a give Number

36. Write a c program to create a new data type Date with the help of structure and typedef. Also write following user defined function for date manipulation.
    a) To return next Date,
    b) To return next Month,
    c) To return next Year,
    d) To add few Days in a date
    e) To add few Months in a date
    f) To return the date of the week of a given date.

g) To return Month name from the date.
h) To Display the Date in various format as: Date Display(Date d1, char *format)

Here Date is the newly created data type. The format string can hold the following values: “DDMMYYYY”, “MMDDYY”, “MON, DD, YYYY”

37. Write a C program to implement myprintf and myscanf functions using concept of variable number of arguments. (using getch, putch, gets and puts function)
38. Write a C program that creates an Employee text file. Records are empid, empname, designation, qualification, salary, experience, Research work, address, city phone.
39. Write C programs for the following operation to work like DOS Commands:
    a) type abc.txt
    b) copy source1.txt source2.txt
    c) compare source1.txt source2.txt
    d) concatenate source1.txt source2.txt
40. Write a C program to open two files containing integers (in sorted order) and merge their contents.
41. Write a C program to count the number of vowels, consonants, digits, spaces, other symbols, words and lines in a given text file.
42. Write C code to check if an integer is a power of 2 or not
43. Write a C program to set a particular bit in a given number.
44. Write a C program to reset a particular bit in a given number.

CS - 103 Operating Systems

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	Introduction: Evolution of operating systems, operating system concepts Processor Management: Concepts, Algorithms for batch processing Processor Management Algorithms for time sharing operating systems, Introduction to real time systems.
Unit 2	Memory Management: Concepts, single user memory management. Partition memory allocation Virtual memory management using paging and segmentation techniques.
Unit 3	Concurrent Processes: Mutual exclusion and synchronization Techniques of inter process Deadlock handling.
Unit 4	File Management: Operations on a file, structure of a file System Free block list, keeping track of blocks allocated to a file, directory structure, sharing and protection of files, file system Reliability, Unix file system Device Management: Goals of input/output software design, Structure of device hardware and software.
Unit 5	Layers of I/O software, structure of device drivers, Disk driver, disk arm scheduling algorithms, terminal driver, clock driver etc. Introduction to network and distributed operating systems.

1. Text Book :
   Operating System Concepts, Addison Wesley, 4th Edition, A. Silberschatz and P. Galvin. 1994.Reference books :
2. Reference books :
   1. Design of Unix operating system, Bach M., Pearson Education.
   2. Operating systems, 4rth Edition, William Stallings, Pearson Education, 2003.

CS - 104 Discrete Structures

Max. Marks : 50 **Min. Marks : 17**

Unit No.	.Topics
Unit 1	**The Foundations:** Logic, Sets and Functions: Introduction to set theory, mathematical logic, prepositions, prepositional equivalences, predicates and quantifiers. Importance of Quantifiers. The Foundations: Logic, Sets and Functions: Sets, set operations, fuzzy sets, functions, functions for computer science, sequences and summations. **Mathematical reasoning:** Introduction to Methods of proof, mathematical induction. Use of mathematical induction to solve different problems. Importance of recursions in computer science, scope of recursions, Recursive definitions, recursive algorithms.
Unit 2	**Combinatorics:** The basics of counting, The sum rule, The product rule, The Pigeonhole Principle, Permutations with repetitions, Permutations without repetitions, Circular Permutations. Applications of combinations. Applications of Combinatorics to solve Committee problems, word problems, puzzle problems etc. Applications of Combinatorics to understand Telephone numbering plan, understanding Internet addresses, Advanced counting techniques, recurrence relations, solving recurrence relations, algorithm design, Basic understanding of complexities, basic problems of complexity of algorithms.
Unit 3	**Relations:** Relation definition , Importance of relations in computer science, Relations and their properties, Unary relations , Binary relations, Ternary relations , n-ary relations and their applications, closures of relations, equivalence relations, partial ordering. Representing relations, relation matrix, relation graph, composite relation. Operations on relations – union, intersection and join. Concepts of least upper bond, Greatest lower bond, maximal element, minimal element, Greatest element, Least element of a partially ordered set, lattices, sub lattices, chains and antichains.
Unit 4	**Graphs:** Introduction to Graphs, Importance of graph theory in computer science, Graph terminology, representing graphs, graph types, graph models, and graph isomorphism. Connectivity, Euler and Hamiltonian Paths, shortest path problems, planar graphs, graph colouring, chromatic number, Euler’s formula, kuratowski’s theorem. The four colour problem, Applications of Graph Colouring, Introduction to Trees, applications of trees, tree traversal, trees and sorting, Spanning trees, minimum spanning trees.
Unit 5	**Languages and Grammars:** Introduction to Languages and Grammars, solving problems for validity of statements according to the grammar. Importance of Language theory in Computer Science, Importance of Derivation trees, solving problems of Derivation trees, Importance of Parsing, Phrase-Structure Grammars, Types of Phrase structure grammars.

1. Text Book :

   Kenneth H. Rosen “Discrete Mathematics and its Applications” , 5th edition , Tata McGraw-Hill Edition.Let us C, Yashavant Kanetkar, BPB Publications.
2. Reference books :
   1. Kolman, Busby & Ross "Discrete Mathematical Structures" ,5th edition Pearson Education
   2. Narsingh Deo "Graph Theory with Applications to Engineering. & Computer Science", 4th edition, Prentice Hall of India .
   3. Discrete Structures, Logic and Computability by James L. Hein, 2nd edition, Narosa Publishing House.
   4. Discrete and Combinatorial Mathematics, 5th Edition by Ralph P. Grimaldi, Addison-Wesley publication.

Class Assignments:

1. Find a formula for the sum of the first *n* even positive integers.
2. Use mathematical induction to prove the formula that you found in Exercise 1.
3. Use mathematical induction to prove that 3 + 3 • 5 +
4. 5_2 +-----+ 3-5" = 3(5"+' - l)/4 whenever *n* is a nonnegative integer.
5. Use mathematical induction to prove that 2 — 2 . 7 + 2.7_2------+ 2(-7)" = (1 - (-7)"$_{+1}$)/4 whenever *n* is a nonnegative integer.
6. Find a formula for ½ +1/4 + 1/8 +……….+1/$2_{\mathbf{n}}$ by examining the values of this expression for small values of n. Use mathematical induction to prove your result.
7. Show that l_2 + 2_2 + • • • + n_2= n(n + l)(2n + l)/6 whenever *n* is a positive integer.
8. Show **that l_3 + 2_3 + • • • + *n_3* = *[n(n* + l)/2]$_2$** whenever *n* is a positive integer.
9. Prove that **l_2 + 3_2 + 5_2 + ….. + (2n + l)$_2$ = (n + 1) *(n* + l)(2n + 3)/3** whenever *n* is a nonnegative integer.
10. Prove that 1 • 1! + 2 . 2! +----+ -*n* . *n! = (n +1)! – 1* whenever *n* is a positive integer.
11. Show by mathematical induction that if *h* > — l , then 1 + *nh* <= (1 + *h)*$_{\mathbf{n}}$ for all nonnegative integers *n.* This is called **Bernoulli's inequality.**
12. Prove that 3" < *n!* whenever *n* is a positive integer greater than 6.
13. Show that 2_n > n_2 whenever *n* is an integer greater than 4.
14. Use mathematical induction to prove that ***n! < n*** $_{\mathbf{n}}$ whenever *n* is a positive integer greater than 1.
15. Prove using mathematical induction that 1 -2 + 2-3 +………..+ n(n+1) = *n(n* + l)(n+2)/3
16. Whenever n is a positive integer.
17. Use mathematical induction to show that 3 divides n $_3$ + *2n* whenever *n* is a nonnegative integer.
18. Use mathematical induction to show that 5 divides *n* $_5$ — *n* whenever *n* is a nonnegative integer.
19. Use mathematical induction to show that 6 divides *n* $_3$ — *n* whenever *n* is a nonnegative integer.
20. Use mathematical induction to show that n_2 — 1 is divisible by 8 whenever *n* is an odd positive integer.
21. Use mathematical induction to show that n_2 — *7n* + 12 is nonnegative if *n* is an integer greater than 3.
22. Use mathematical induction to prove that a set with *n* elements has *n(n* — l)/2 subsets containing exactly two elements whenever *n* is an integer greater than or equal to 2.
23. Use mathematical induction to prove that a set with *n* elements has *n(n — \)(n —* 2)/6 subsets containing exactly three elements whenever *n* is an integer greater than or equal to 3.
24. Write a small procedure for addition of a node in a binary search tree.
25. Define the following terms :-
    a) Graph Isomorphism
    b) Map coloring
    c) Chromatic Polynomial
    d) Phrase structure grammar
    e) Euler graph.
    f) Bipartite Graph
    g) Hamiltonian circuits
26. Write a small C code to create a digraph.
27. Write a small C code for insertion of a node in binary search tree.
28. Implement Topological sorting algorithm in C language.

29. Write a C code to decide whether a graph is Bipartite.
30. Write a small C code to create a graph and find its adjacency matrix.
31. Write a small C code to create a graph and find its adjacency list.Write a small C code to create a graph and find its incidence matrix.
32. Implement Dijstra’s algorithm in C.
33. Write a small C code to determine isomorphism of two graphs.
34. Write a small C code to determine whether a graph is euler graph.
35. Write a small C code to determine whether a graph has Hamilton path.
36. Implement Prim’s algorithm in C for finding minimum spanning tree.
37. Implement Kruskal’s algorithm in C for finding minimum spanning tree.
38. Write a small C code to create a graph and find its subgraphs.
39. Write a small C code for breadth first traversal.

IC-105 Communication Skills

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Fundaments of Communication (OHP & PPP):** Definitions, importance, forms of communication, process of communication, channels, barriers and strategies to overcome barriers of communication. **Listening (PPP):** Def, Importance, Benefits, barriers, approaches, be a better listener, exercises and cases.
Unit 2	**Advance Communication:** Why communication? Art of communication, V3 communication, Key elements of IP communication, Quizzes, exercises and cases / incidents for practice.
Unit 3	**Group Discussions:(PPP)** Definitions, importance, process, points to be borne in mind while participating, Dos and Don'ts. Practice- if time permits or to be covered in PDP. **Interview (PPP)** Types of Interviews, Points to be borne in mind as an interviewer or an Interviewee, commonly asked questions, Dos and Don'ts. Practice- if time permits or to be covered in PDP.
Unit 4	**Transactional Analysis: (PPP)** Transactional analysis, Johari Window, FIRO-B (PPP) **Written Communication:** Cover letter, Report writing, documentation, business correspondence, preparation of manuals and project reports, Drafting emails.
Unit 5	**Negotiation Skill**: Basic principles, Building understanding, process of negotiation, essentials of negotiations. Contemporary Communication Styles, technology enabled communication

1. Text Book :
   Chturvedi, P.D. and Chaturvedi Mukesh (2004), "Business Communication" Pearson Education, Singapore Pvt. Ltd.
2. Reference books :
   1. Business Communication by ICMR, Feb 2001.
   2. Toropov Brandon (2000), "Last Minute Interview Tips", Jaico Publishing House, Mumbai.
   3. Heller Robert (1998), "Essential DK Managers: Communication Clearly", Dorling Kindersley, London.

Class Assignments:

1. Making their Resumes.
2. Writing formal letters, Memos, drafting emails, notices.
3. Create a questionnaire, and meet new people they never met and know about them.

CS - 201 Data Structures using C++

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Introduction to C++:** Structures, Variables in C++, References, Functions, Function Overloading, Default Values for Formal Arguments of Functions, Inline Functions. **Class and Objects**: Introduction to Classes and Objects Constructors, destructors, friend function, dynamic memory allocation, Inheritance, Overloading, Polymorphism, Templates.
Unit 2	Definition of data structures and abstract data types. Examples and real life applications **Data Structures**: Arrays, Stacks, Queues, Dequeues
Unit 3	Linked Lists, Singly and Doubly linked list Applications of Linked Lists, Linked implementation of queue and stacks **Binary Search Tree :** Definition and Implementation- preorder, post order, inorder traversal, Red Black Tree Definition and implementation
Unit 4	**Graphs:** Definition and implementation, Hash function, Collision Resolution Techniques, Hashing Applications, Time Complexity, Big – Oh - notation, Running Times, Best Case, Worst Case, Average Case, Factors depends on running time, Introduction to Recursion, Divide and Conquer Algorithm, Evaluating time Complexity
Unit 5	Straight Sequential Search, Array implementations, Linked implementation, Binary Search, Interpolation Search. Introduction, Sorting by exchange, selection, insertions, Bubble sort, Selection sort, Insertion sort, Efficiency of above algorithms, Merge sort, Quick sort Algorithm, Heap sort, Heap Construction, Heap sort, Radix sort.

1. Text Book :
   Data Structures using C by A. M. Tenenbaum, Langsam, Moshe J. Augentem, PHI Pub.
2. Reference books :
   1. Data Structures and Algorithms by A.V. Aho, J.E. Hopcroft and T.D. Ullman, Original edition, Addison-Wesley, 1999, Low Priced Edition.
   2. Fundamentals of Data structures by Ellis Horowitz & Sartaj Sahni, Pub, 1983,AW.

Class Assignments:

1. Write a program to display message:

   Welcome to C++
   Wish you all the best.
2. Write a program to find the area of circle, rectangle or triangle.
3. Write a program that allows the user to enter the number, and then generate the table, formatting it into ten columns and 20 lines. Interaction with the program should look like this.

```
          Enter a number: 2
2     4     6     8     10    12    14    16    18    20
22    24    26    28    30    32    34    36    38    40
42    44    46    48    50    52    54    56    58    60
```

4. Write a temperature – conversion program that gives the user the option of converting Fahrenheit to Celsius or Celsius to Fahrenheit. Then carry out the conversion. Use Floating point numbers.
5. Create the equivalent of four-function calculator. The program should request the user to enter a number, an operator, and another number. It should then carry out the specified arithmetical operation; adding, subtracting, multiplying or dividing the two numbers. (It should use a switch statement to select the operation). Finally it should display the result. When it finishes the calculation, the program should ask if the user want to do another calculation. The response can be ‘y’ or ‘n’.
6. A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767), and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers.
7. A point on the two-dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. For example, (4,5) represents a point 4 units to the right of the origin along the X axis, and 5 units up the Y axis. The sum of two points can be defined as new points, and whose Y coordinate is the sum of their Y coordinates.
8. Write a program that uses a structure called point to model A point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point.
9. Create a structure called volume that uses three variables of type Distance to model the volume of a room. Initialize a variable of type Volume to specific dimensions, and then calculate the volume it represents, and print out the result. To calculate the volume, convert each dimension from a Distance variable to a variable of type float representing feet and fractions of a foot, and then multiply the resulting three numbers.
10. Write a function called circarea ( ) that finds the area of a circle in a similar way. It should take an argument of type float and return an argument of the same type. Write a main ( ) function that gets a radius value from the user, calls cIrcarea ( ), and displays the result.
11. Raising a number n to a power p is the same as multiplying n by itself p times. Write a function called power ( ) that takes double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is committed, the number will be squared. Write a main ( ) function that gets values from the user to test this function.
12. Write a function called zero Smaller ( ) that is passed two int argument by reference and then sets the smaller of the two numbers to 0. Write a main ( ) program to exercise this function.
13. Write a function that takes two Distance values as arguments and returns the larger one. Include a main ( ) program that accepts two Distance figures from the user, compares them, and displays the larger.
14. Create a class that imitates part of the functionality of the basis data type int, Call the class int (note different spelling). The only data in this class is an int variable. Include member functions to initialize an Int to 0, to initialize it to an int value, to display it (it looks just like an int), and to add two Int values.
15. Write a program that exercises this class by creating two initialized and one un-initialized int values, adding these two initialized values and placing the response in the un-initialized value, and then displaying this result.
16. Create a class called time that has separate int member data for hours, minutes, and second. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it, in 11:59:59 format. The final member

function should add two objects of type time passed as arguments. A main ( ) program should create two initialized time objects, and one that isn't initialized. Then it should add the two initialized values together, leaving the result in the third time variable. Finally it should display the value of this third variable.

17. Write a function called reversit ( ) that reverse a string (anarray of char). Use a for loop that swaps the first and last characters, then the second and next-to-last characters, and so on. The string should be passed to reverse it ( ) as an argument.Write a program to exercise reversit the program should get a string from the user, call reversit ( ) and print out the result. Use an input method that allows embedded blanks. Test the program with Napoleon's famous phrase, "Able was I ere I saw Elba".

18. Create a class called employee that contains a name (an array of char) and an employee number (type Long). Include a member function called get data ( ) to get data from the user for insertion into the object, and another function called put data ( ) to display the data. Assume the name has no embedded blanks.Write a main () program use this class. It should create an array of type employee, and then the user input data for up to 10 employees. Finally, it should print out the data for all the employees.

19. Write a program that substitutes an overloaded += operator for the overloaded+ operator in the STRPLUS program. This operator should allow statements likes1 +=s2; where s2 is added (concentrated) to s1 and the result left in s1. The operator should also permit the results of the operation to be used in other calculations, as in s3 = s1 +=s2.

20. Create a class int. Overload all five integer arithmetic operators (+; =; *; /; and %) so that they operate an objects of type int. If the result of any such arithmetic operation exceeds the normal range of ints = from = 32,768 to 32;767 –have the operator print a warning and terminate the program. Such a data type might be useful where mistakes caused by arithmetic overflow are unacceptable. Hint: To facilitate checking for overflow; perform the calculations using type long. Write a program to test this class.

21. Imagine a publishing company that markets both book and audio-cassette versions of its works. Create a class publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes; book, which adds a page count (type int); and tape, which adds a playing time in minutes (type float). Each of these three classes should have a get data () function to get its data from the user at the keyboard, and a put data ( ) function to display its data.Write a main ( ) program to test the book and tape classes by creating instances of them, asking the user to fill in their data with get data ( ), and then displaying the data with put data ( ).

22. Write a program that reads a group of numbers from the user and places them in an array of type float. Once the numbers are stored in the array, the program should average them and print the result. Use pointer notation wherever possible.

23. Start with the String class from the NEWSTR. Add a member function called up it ( ) that converts the string to all upper case. You can use the toupper ( ) library function, which takes a single character as an argument and return a character that has been converted (if necessary) to uppercase. This function uses the CTYPE,h H header file. Write some code in main ( ) to test this function.

24. Add a destructor to the LINKLIST program. It should delete all the links when a link list object is destroyed. It can do this by following along the chain, deleting each links as it goes. You can test the destructor by having it display a message each time it deletes a link; it should delete the same number of links that were added to the list. (A destructor is called automatically by the system for any existing objects when the program exits.)

25. Imagine the same publishing company that markets both book and audio-cassette versions of its works. As in that exercise, create a class called publication that stores the title (a string) and price (type float) of a publication. From this class derive two classes: book, which adds a page count (type int); and tape, which adds a playing time in minutes (type

float). Each of the three classes should have a get data ( ) function to get its data from the user at the keyboard, and a put data ( ) function to display the data.

26. Write a C++ Program to construct a Class stack of integers and to perform
27. the following operations on it:
    a) Push
    b) Pop
    c) Display
28. The program should print appropriate messages for stack overflow, stack underflow, and stack empty.
29. Write a C++ Program to simulate the working of a queue of integers
30. using an array. Provide the following operations:
    a) Insert
    b) Delete
    c) Display
31. Write a C++ Program to simulate the working of a circular queue of integers using an array. Provide the following operations:
    a) Insert
    b) Delete
    c) Display
32. Write a C++ Program to convert and print a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and the binary operators + (plus), - (minus), * (multiply) and / (divide).
33. Write a C++ Program to evaluate a valid suffix/postfix expression using stack. Assume that the suffix/postfix expression is read as a single line consisting of non-negative single digit operands and binary arithmetic operators. The arithmetic operators are + (add), - (subtract), * (multiply) and / (divide).
34. Write a C++ Program using dynamic variables and pointers, to construct a singly linked list consisting of the following information in each node: student id (integer), student name (character string) and semester (integer). The operations to be supported are:
    a) The insertion operation
        I. At the front of a list
        II. At the back of the list
        III. At any position in the list
    b) Deleting a node based on student id. If the specified node is not present in the list an error message should be displayed. Both the options should be demonstrated.
    c) Searching a node based on student id and update the information content. If the specified node is not present in the list an error message should be displayed. Both situations should be displayed.
    d) Displaying all the nodes in the list
35. Write a C++ Program using dynamic variables and pointers to construct a stack of integers using singly linked list and to perform the following operations:
    a) Push
    b) Pop
    c) Display
36. The program should print appropriate messages for stack overflow and stack empty.
37. Write a C++ program using dynamic variables and pointers to construct a queue of integers using singly linked list and to perform the following operations:
    a) Insert
    b) Delete
    c) Display

38. Write a C++ Program to support the following operations on a doubly linked list where each node consists of integers:
    a) Create a doubly linked list by adding each node at the front.
    b) Insert a new node to the left of the node whose key value is read as an input
    c) Delete the node of a given data, if it is found, otherwise display appropriate message.
    d) Display the contents of the list.
39. Write a C++ Program
    a) To construct a binary search tree of integers.
    b) To traverse the tree using all the methods i.e., inorder,
    c) Preorder and postorder.
    d) To display the elements in the tree.
40. Write a C++ program to implement multiway search trees.
41. Write a C++ program to implement AVL trees
42. Write recursive C++ Programs for
    a) Searching an element on a given list of integers using the Binary Search method.
    b) Solving the Towers of Hanoi problem.
43. Write a C++ program to implement Hashing Functions.

CS - 202 Database Management System

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Introduction:** Advantages of DBMS approach, Various views of data, data independence, schema & sub-schema, primary concept of data models, database languages, transaction management, database administrator &user, data dictionary, database architectures. **ER model:** Basic concept, design issues, mapping constraint, keys, ER diagram, weak & strong entity-sets, specialization & generalization, aggregation, inheritance, design of ER schema, Reduction of ER Schema to tables. Domains, relation, kind of relation, Relational databases, Various types of keys: candidate, primary, alternate & foreign keys.
Unit 2	**Relational Algebra and SQL:** The structure, relational algebra with extended operations, modification of database, Idea of relational calculus. **Relational Database:** Basic structure of SQL, Set operation, Aggregate functions, Null values, Nested Sub queries, derived relations, views, Modification of database, join relation, Domain, relation & keys, DDL in SQL. **Programming concepts of PL/SQL, Stored procedure, Database connectivity with ODBC/JDBC.**
Unit 3	**Functional dependencies:** Basic definitions, Trivial & non trivial dependencies, closure set of dependencies & of attributes, Irreducible set of dependencies, FD diagram**.** **Normalization:** Introduction to normalization, non loss decomposition, First, second and third normal forms, dependency preservation, BCNF, multivalue dependencies and fourth normal form, join dependencies and fifth normal form. **Database Integrity:** general idea, integrity rules, Domain rules, Attributes rules, assertion, triggers, integrity & SQL
Unit 4	**Transaction Management:** basic concept, ACID properties, transaction state, Implementation of atomicity & durability, Concurrent execution, Basic idea of serializability. **Concurrency & Recovery:** Basic idea of concurrency control, basic idea of deadlock, Failure Classification, storage structure-types, stable storage implementation, data access, recovery & Atomicity: log based recovery, deferred database modification, immediate database modification, checkpoints.
Unit 5	**Storage Structure:** overview of physical storage media, magnetic disk: performance & optimization, RAID. **File Organization:** File organization, Organization of records in files, basic concept of Indexing, ordered indices: B+ tree & B tree index files. **Query processing, Query optimization,** I**ntroduction to data mining and data warehousing.**

1. Text Book :
   "Database System concepts – Henry F. Korth, Tata McGraw Hill" 4th Edition
2. Reference books :
   1. **"**Fundamentals of Database Systems", Elmasri R, Navathe S, Addison Wesley 4th Ed., ISBN 0321122267
   2. An introduction to database system- Bipin C. Desai
   3. An introduction to Database System - C.J Date
   4. SQL, PL/SQL The programming language of Oracle- Ivan Bayross

Effective for Students Admitted in July, 2011 onwards

Class Assignments:

(Student should submit their assignment in hard copy (hand written) as well as soft copy to the respective faculty)

A. Prepare the following case study on ER diagram and normalized database design based on FD's

(a) Database design for Retail Banking
(b) Database design for Technical Training Institute
(c) Database design for an Internet Book Shop
(d) Database design for Customer Order Warehouse
(e) Database design for University Registrar's Office

B. Define the schema for the following databases with specific data type and constraints, the table name and its fields name are to be taken from database description which are given below :

1. Sales Information System

A database is being constructed for storing sales information system. A product can be described with a unique product number, product name, selling price, manufacturer name. The product can be sold to a particular client and each client have it own unique client number, client name, client addresses, city, pin code, state and total balance to be required to paid. Each client orders to buy product from the salesman. In the order, it has unique sales order number, sales order date, client number, salesman number (unique), billed whole payment by the party or not and its delivery date. The information associated with salesman is name, addresses, city, pin code, state, salary of the sales man, delivery date, total quantity ordered, product rate.

Write the SQL queries for the following –

a) Retrieve the list of names and the cities of all the clients.
b) List the various products available.
c) Find the names of all clients having 'a' as the second letter in their names.
d) List all the clients who are located in 'INDORE'.
e) Find the products whose selling price is greater than 2000 and less than or equal to 5000.
f) Add a new column NEW_PRICE into the product_master table.
g) Rename the column product_rate of Sales_Order_Details to new_product_rate.
h) List the products in sorted order of their description.
i) Display the order number and date on which the clients placed their order.
j) Delete all the records having delivery date before 25th March, 2010.
k) Change the delivery date of order number ON01008 to 16-05-10.
l) Change the bal_due of client_no CN01003 to 1200.
m) Find the product with description as 'HDD1034' and 'DVDRW'.
n) List the names, city and state of the clients not in the state of 'MP'.
o) List of all orders that were cancelled in the month of March.

2. College Department Management

A student is described by a unique Roll Number, Name, Address, and Semester. Each student enrols himself/herself in an Academic programme offered by a Department. Academic programmes have programme name(unique), duration, a programme code(unique) and a list of subjects (both core and elective subject) while the departments have department code (unique), department name (unique), HoD who is a Teacher and list of courses offered by it. Each teacher is described by employee code (unique), name, department and designation. A student registers some courses in a semester. A course is described by a unique course number, title of the course, credit allotted for the course and offering department. Database stores the grades obtained by different students in different courses registered by him/her in different semesters. Database also stores information about the courses offered by a department in a semester, the corresponding teacher(s) for each course.

Write the SQL queries for the following –

a) Find all the students' name, city, course allotted from the SCS department.
b) List the total number of Faculty in the SCS department.
c) List the available courses from the SCS department.
d) List the all students in a particular semester.
e) List the students who earned CGPA greater than or equal to 8.5.
f) How much subjects are registered by a student in each semester.
g) List the common students who are allotted the same courses of both the programme MCA and M. Tech.
h) List the total number of student enrolled in the subject DBMS.
i) Retrieve the semester of the student under DBMS subject.
j) Retrieve the entire student name and arrange into ascending order.
k) Modify a student address DEWAS to INDORE where sdt_id='CSI08002'.
l) Find the total credit point of student required to complete for a course like MCA.
m) List the all courses which are related to computer science.
n) Retrieve all the students located at 'INDORE'.
o) Find the total number of department in our database.
p) List the all courses which are related to computer science.

3. Bank Database System

A bank database keeps record of the details of customers, accounts, loans and transactions such as deposits or withdraws. Customer record includes customer id, customer name, address, age, contact number, email id etc., accounts details are account number, account type(fixed account, savings account, monthly account etc), date of creation of the account. Transaction detail keeps information about amount deposited or withdrawn to/from a particular account and the date of transaction. The database should also store record of loans which include loan amount, loan date and the account number to which the loan is granted.

Make appropriate tables for the above database and try to find out the following queries:

a) List the details of account holders who have a 'savings' account.
b) List the Name and address of account holders with loan amount more than 50,000.
c) Change the name of the customer to 'ABC' whose account number is 'TU001'
d) List the account number with total deposit more than 80,000.
e) List the number of fixed deposit accounts in the bank.
f) Display the details of customers who created their accounts between '20-jan-10' to '20-Mar-10'.
g) Display the detailed transactions on 28th Feb, 2010.
h) Display the total amount deposited and withdrawn on 29th Aug, 2008.
i) List the details of customers who have a loan.

4. Library Information System

A library database stores information about books, journals, magazines etc. Searching for books can be done by author, title, subjects etc. Similarly journals can be searched by subject area, publisher etc. It should also be possible to see which book is issued to which student and belonging department.

Write the queries for the following –

a) List the names of the books issued between 21-Jan-10 and 29-Jan-10.
b) Retrieve the name and number of books by a particular author.
c) Retrieve the name of the publisher which has maximum number of books.
d) Count the total number of books in the library.
e) Count the number of books issued to a student with Roll no ‘CSB06001’.
f) Change the author of the book to ‘ABC’ with book id=’BK003’.
g) Retrieve the name of the student to whom the book named ‘Database System’, by ‘E. Navathe’ is issued.
h) Display the total number of books issued to different departments.
i) List the name of the books where subject is like ‘ora’.

5. Hospital Information System

Hospital information system stores following information:

Patients - indoor/outdoor, medicines/lab tests (including results) prescribed to patients, information if a patient if referred to other expert/hospital. Doctors - specialization, patients attended etc. Different wards/beds and patients allotted to them etc. Patient registration form contains Registration number, Patient name, Address, Gender, Bed number, date of registration, refer doctor id etc. Doctor information contains Doctor code, Doctor Name, Specialization etc. Lab test information contains Test name, test number, test date, results and referred doctor’s code. Bed information contains bed number, ward number and status (whether allotted or not).

Queries:

a) Display the details of patients admitted between ‘20-jan-10’ and ’20-Mar-10’.
b) Change the name of the patient to ‘Ram’ whose patient id=’PT011’.
c) Display the names of the patients and lab test results performed on ‘20-jan-10’.
d) Display the number of patients taking treatment under doctor =’ABC’.
e) Retrieve the name of doctor who is taking care of maximum number of patients.
f) Change the bed number of the patient to 456 where patient id=’PT023’.
g) Change the status of bed with bed number 123 with ‘not allotted’.
h) List the bed details which are free in ward number 10.
i) List the name of male patients in ward no 13 taking treatment under doctor ‘XYZ’.
j) List the details of patients with age more than 50 taking treatment under a doctor, whose name like ‘das’.

6. Payroll System of DAVV University

Write the queries for the following –

a) List all the employees of SCS department.
b) Retrieve all the employees who have the gross salary greater than or equal to Rs-21,000.
c) Find the DA, TA, HRA of the employee name ‘Nitin’.
d) Find the total leave get by an employee.
e) List the employees department.
f) List the name grades of the employee of the SCS department.
g) Find how much tax will be required to pay by an employee in each month.
h) List the employees who are joined between ’10-APR-07’ to ’28-AUG-08’.
i) Retrieve the mail id of all employees of SCS department.
j) Total leave available of the employee ‘Ravindra’.
k) Add a new column ‘employee phone’ to a table employee.
l) List the employees with basic salary 8000.
m) Find the employees who have the highest bank balance.
n) Retrieve the employees who have the bank loan.
o) Find the bank balance of the employee ‘SCSMP07001’.

CS - 203 Software Engineering

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Introduction to Software Engineering & Software Processes:** Software problem, Software engineering problem, Software engineering approach, Software characteristics and Applications. **Software processes** and its components, characteristics of software processes, Software development processes: Linear Sequential model, Prototyping model, Iterative Enhancement model, Spiral model, Time boxing model, RAD model, Component based development, Comparative study of various development models.
Unit 2	**Project management process:** The people, product, process and project, Phases of project management process, Project life cycle, the W5HH principle. Software configuration management process, Process management process. Project Planning: Metrics and measurements, Project estimation (Size & Cost), Project Scheduling, Staffing and personnel planning, Software configuration management plans, Quality assurance plans, Project monitoring plans, Risk management.
Unit 3	**Software Requirement Analysis and Specification:** Software requirements, Functional & Non functional requirements, Problem analysis (Structured analysis and Object Oriented analysis, Prototyping approach). Software Requirements specifications (SRS), Validation and Verification, Metrics. **Software Design:** Design principles: Problem partitioning and hierarchy, Abstraction, Modularity, Top-down and Bottom-up strategies. Effective Modular design: functional independency, Cohesion, Coupling. Structured design methodology.
Unit 4	**Software Quality Assurance**: Quality concept, Quality management system, movements and assurance, Software reviews: formal and technical. Formal approaches to SQA, Statistical software quality assurance, Software reliability, Capability Maturity Model (CMM), ISO 9000, Six sigma, SQA plan.
Unit 5	**Software Testing**: Software testing techniques: Testing fundamentals, White box testing, Black box testing, Testing for specialized environments, architectures and applications. Software testing strategies: A strategic approach to software testing, Strategic issues, Unit testing, Integration testing, Validation testing and system testing, The art of debugging.

1. Text Book :

   An Integrated Approach to Software Engineering- Pankaj Jalote, Narosa Publishing House.

2. Reference books :

   1. Software Engineering- Ian Sommerville, Pearson Education, New Delhi.
   2. Software Engineering Concepts-Richard E. Fairly, Tata McGraw Hill Inc. New York.
   3. Software Engineering: Principle & Practice-W. S. Jawadekar, Tata McGraw-Hill, New York.

Class Assignments:

Assignment 1:

Select a project that you want to develop and define the aim, objectives, and goals of your project. Select a software development methodology to develop the same project. Perform various phases of the methodology to produce work-products.

Assignment 2:

1. Suppose a program for solving a problem costs C, and a programming product for solving that problem costs 9C. Where do you think the extra 8C cost is spent? Suggest a possible breakdown of this extra cost.
2. If you are given extra time to improve the reliability of the final product developing a software product, how would you distribute this extra time?
3. Suggest ways of reducing the cost due to rework.
4. Suggest some ways to detect software errors in the early phases (when implementation is not yet complete.

Assignment 3:

1. Define & discuss Time boxing and Component based development model.
2. What problems will a Software Development Organization face if it does not have sufficiently document in its software process?
3. Suppose you were to plan to undertake the development of product base with large number of technical as well as customer, justify which life cycle model would you adopt.
4. Suggest a suitable life cycle model, for software project which your organization has undertaken on behalf of a certain change his requirements frequently justify.
5. Which of the following process models discussed in this chapter would you follow for the following projects? Give justifications.

a) A simple data processing project
b) A data entry system for office staff who have never used computers before. The user interface and user friendliness are extremely important.
c) A new system for comparing fingerprints. It is not clear if the current algorithms can compare fingerprints in the given response time constraints.
d) A spreadsheet system that has some basic features and many other desirable features that use these basic features
e) A new missile tracking system. It is not known if the current hardware /software technology is mature enough to achieve the goals.
f) An on-line inventory management system for an automobile industry.
g) A flight control system with extremely high reliability. There are many potential hazards with such a system.

5. It is reasonable to assume that if software is easy to test, it will be easy to maintain. Suppose that by putting extra effort in design and coding you increase the cost of these phases by 15%, but you reduce the cost of testing and maintenance by 5%. Will you put in the extra effort?
6. Suppose you can measure the number the defects detected during the various reviews and testing. However, the customer requires an estimate of the Number of defects remaining at

delivery time. How will you build a model to predict this? Assume the existence of any data you need.

Assignment 4:

1. A database system is to be developed. After the requirements, its size is estimated to be 10,000 lines of code. Estimate the overall cost using the Watson and Felix model.
2. Consider a project to develop a full-screen editor. The major components identified are (1) screen edit, (2) command language interpreter, (3) file input and output, (4) cursor movement, and (5) screen movement. The sizes for these are estimated to be 4K, 2K, 1K, 2K and 3k delivered source code lines. Use the COCOMO model to determine (a) overall cost and schedule estimates (assumes values for different from 1.0),(b)cost and schedule estimates for different phases, and (c) detailed cost and schedule estimates for the different components.
3. Suppose each communication path between two people consumes 5% of each person's time. For a project that requires 12 staff-months of programming work, how many people will be needed to finish the project in four months if
(a) The democratic team structure is used, and
(b) The chief-programmer team structure is used? If the team consists of four persons, what is the difference in the completion time for a team using the democratic structure and a team using the chief-programmer structure?
4. Assume that testing (and bug fixing) effort is proportional to the number of errors detected (regardless of the nature of errors). Suppose that testing detects 90% of the total errors in the SW (10% remain undetected). By adding design and code reviews, suppose the cost of the design and coding phases increases by 10% each(from the base distribution given earlier),and 10% of the errors are detected in design reviews and 10% in code reviews .(So, testing now detects only 70% of errors.)What is the impact on the overall cost of reviews.

Assignment 5:

1. Suppose you are developing a project in an organic mode. you have estimated the size of the product is about 100000 LOC. Compute normal effort and development time.
2. Suppose that an on-the-shelf s/w product for business applicant coast 15000/- and that its size is 40 KLOC. Assuming that in house engineer coast 6000/- per program month including overheads, would it be more effective by the product build it, which element of the cost are not included in the COCOMO MODEL. What additional factor should be considered making the buy and build decision.
3. The industry average productivity figure for engineers is only 10 LOC per day .What is the reason for the low productivity. Can be attributing the poor programming skills of the Software engineers.
4. As a project manager identify the traits that you would look for software engineers .While trying to select person for your team.
5. For the same number of lines of code and same development team size rank the following software project, in order of their estimated development time
a) A text editor.
b) An employee payroll system.
c) An operating system for a new computer.

Assignment 6:

1. What are central problems in producing the RS for a system?

2. Construct an example of an inconsistent (incomplete) SRS?

3. Make a friend of yours as the client. Perform structured analysis and object oriented analysis for the following problems:

i. An electronic mail system.
ii. A simple student registration system
iii. A system to analyse a person's diet.
iv. A system to manage recipes for a household.
v. A system to fill tax forms for the current year tax laws.

4. A library database contains entries that have the name of the book, followed by the author's name, the publisher's name, and year of publication, ISBN number of the book, and number of copies of the book. Each of the data entries is on a new line. Represent this database as a regular expression.

Assignment 7:

1. What is the cohesion of the following module? How would you change the module to increase cohesion?

```
Procedure file( file_ptr, filr_name, op_name)
Begin
        Case op_name of
        "open" perform activities for opening the file.
        "close" perfor activities for opening the file.
        "print" print the file.
  End case
End.
```

2. What is the difference between a flow chart and a structure chart?

3. Draw the structure chart for the following program:

```
Main();
{
        Int x, y;
        X=0; y=0;
        A(); b();}
A()
{x=x+y; y=y+5;}
B()
{x=x+5; y=y+x; a();}
```

How would you modify this program to improve the modularity?

4. Use the structured design methodology to produce a design for the following :

i. A system to convert ASCII to EBSDIC?
ii. A system that acts as a calculator with only basic arithmetic functions?
iii. A system do do student registration in the manner it is done at your college?
iv. A system to manage the inventory at a hardware store?

5. Given a structure with high fan-out, how would you convert it to a structure with a low fan-out?

Assignment 8:

1. Define error, fault, and failure? What is the difference between a fault and a failure? Does testing observe faults or failures?
2. What are the different levels of testing and the goals of different levels?
3. What is the goal of testing? Why the psychology of tester is is important

CS - 204 Computer Architecture and Organization

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	Technological trends, measuring performance, speed up Amdahl's law. Basic structure of computer hardware: Functional units and components in computer organization: The memory unit, the input and output subsystem, the bus structures, ALU. Program development tools: Compiler, interpreter, and assembler. Introduction to 8085 micro processor architecture. pin out diagram of 8085 microprocessor. Addressing modes of 8085 processor.
Unit 2	Instruction set of 8085 processor: data transfer instructions, arithmetic instructions, branch instructions etc. assembly language programming examples. Introduction to 8088 microprocessor. Bit pair multiplication. Integer division. BCD arithmetic, Design of ALU. Concepts of instruction formats and instruction set, Instruction set types, types of operands and operations. Generation of memory address and addressing modes.
Unit 3	STACKS and QUEUS, GPR organization and stack based organizations. Encoding of machine instructions. Features of RISC and CISC processors. Processing unit design: Processor micro architecture –I Fundamentals concepts for data path implementation. Processor micro architecture-II data path implementation.
Unit 4	Instruction pipelining, instruction pipelining hazards, data dependency hazards and control hazards. Overcoming hazards, instruction set design influence on pipelining. Parallel processing and pipelining, pipelining in RISC and CISC processors. Super scalar processors. In order and out of order execution.
Unit 5	Instruction level parallelism, introduction to VLIW processors, vector processors, CACHES: Data caches, instruction caches and unified caches Cache implementations. Fully associative and direct mapped caches. Write back versus write through caches. **Input Output organization, accessing I/O devices, Interrupts. Memory mapped I/O and I/O mapped I/O.**

1. Text Book :
   Computer Architecture: Scahaum's outlines by Dr. Rajkamal.
2. Reference books :
   1. The 8088 and 8086 Microprocessors by Walter A. Tribel, Avtar Singh.
   2. Computer Organization & Architecture by William Stallings.
   3. Computer Architecture & Parallel Processing, Hwang & Briggs, McGraw Hill
      Microprocessor Archi. Prog. and app. With 8085/8080 By Ramesh S. Gaonkar.

Effective for Students Admitted in July, 2011 onwards

CS - 205 Computer Oriented numerical and statistical method

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Solution of non-linear & transcendental equations** Computer Arithmetic: Floating-point representation of numbers, arithmetic operations with normalized floating-point numbers and their consequences, significant figures. Error in number representation-inherent error, truncation, absolute, relative, percentage and round-off error. Iterative Methods: Bisection method, method of false position, newton-rapson method, secant method, method of successive approximation, concept oriented theoretical consideration of above methods.
Unit 2	**Solution of linear equations** Meaning, conditions for solutions, solution of equation by direct methods - (Gaussian elimination, Gaussian jordan), iterative methods - (Jacobi method, gaussian seidel), ill-conditional equations and solution.
Unit 3	**Interpolation and approximation** Introduction, finite differences, Newton's formulae, Central difference formulae, interpolation with unevenly spaced points, divided difference and their properties, inverse interpolation and double interpolation.
Unit 4	**Numerical integration & solution of ordinary differential equations** Concept of numerical integration with geometrical representation, trapezoidal method, simpson - 1/3 rule, simpson - 3/8 rule, veddle's rule, understanding and solution of Ordinary Differential Equation and theoretical consideration, euler method, modified euler's method, R-K 2nd order & 4th order method, predictor corrector methods.
Unit 5	**Statistics** Graphical representation, Frequency distributions, Measures of central tendency, Measures of dispersions, Correlation, Regression.

TEXT BOOK:

V. Rajaraman, Computer Oriented Numerical Methods, Prentice Hall, India.

REFERENCE BOOKS

1. S. S. Sastry, Introductory Methods of Numerical Analysis.
2. M. K. Jain, S.R.K. Iyengar & R. K. Jain, Numerical Methods for Scientific and Engineering Computation.
3. H. C. Saxena, Finite Differences and Numerical Analysis.
4. Modes A., Numerical Analysis for Computer Science.

Effective for Students Admitted in July, 2011 onwards

CS - 301 Object Oriented Programming Using JAVA

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Introduction to java:** Features of Java, Object-oriented programming overview, Introduction of Java Technologies, How to write simple Java programs, Data Types, Variables, Memory concepts, decision making operators, Naming Conventions, Introduction to Class, Objects, Methods and Instance Variables, Primitive type Vs Reference Type, Initializing Objects with Constructors. Type conversion & casting, Operators, Control Statements(if Single-Selection Statement, if-else Double Selection), while Repetition Statement, for Repetition Statement, do-while Repetition Statements, switch Multiple-Selection Statement, break and continue Statements. Static Method, static field and Math Class, Method Call Stack and Activation Record, Argument Promotion and Casting, Scope of declaration and Method Overloading.
Unit 2	**String Handling:** The String constructors, String operators, Character Exaction, String comparison, String Buffer. **Arrays:** Declaring and Creating Arrays, Enhanced for Statement, Passing Arrays to Method, Multidimensional Arrays, Variable-Length Argument lists, Using Command-line Arguments. final Instance Variables, this reference, static import, overloaded Constructors, Garbage collection and method finalize , Overloading methods, Parameter passing.
Unit 3	**Inheritance:** Extending classes, protected Members, relationship between Superclasses and Subclasses, Using super, Constructor in Subclasses, The Object Class, Object Copying in Java. **Polymorphism:** Method overriding, upcasting, Dynamic Method Dispatch, final Method and classes, Abstract classes and Methods, instanceof operator, Downcasting, Class class, Runtime type Identification **Packages and Interfaces:** Defining a Package, Understanding CLASSPATH, Access Protection, Importing packages, creating own packages. Defining an Interface, Properties of interface, advantages of interface Achieving multiple inheritance through interfaces, Variables in Interfaces, Comparable interface.
Unit 4	**Exception Handling:** Introduction, overview of doing it and keywords used, when to use it, Java Exception Hierarchy, finally block, chained exceptions, declaring new exception types, preconditions and postconditions. **Streams and Files:** Introduction, Data Hierarchy, Files and Streams, Sequential-access Text Files, Object Serialization, Random-Access files, Java Stream class Hierarchy. **Multithreading:** What are threads, The java thread model, Thread priorities, Thread life cycle, Creating thread and executing thread, Thread Synchronization, producer-consumer problem without Synchronization, Producer-consumer problem with Synchronization, Other class and Interfaces in java.util.concurrent, Monitor and Monitor Locks, Thread Groups, Synchronization, Inter-thread Communication.
Unit 5	**Introduction To GUI :** Introduction, Overview of swing Components, Displaying text and Images in a window, Introduction to Event Handling, Common GUI Event Type and Listener Interfaces, How Event Handling Works, Adapter Classes, Layout Managers **Applets:** Applet basics, Applet Architecture, Applet life cycle methods, Applet HTML Tag and attributes, Executing applet in web browser and in the appletviewer, in Passing parameters to Applets, doing GUI programming in applet **Generic and Collection API:** Introduction, Motivation for Generic Methods, Generic Methods : Implementation and Compile- time Translation Issues, Overloading Generic Methods, Generic Classes, Raw Types, Generic and Inheritance **Database connectivity:** JDBC, The design of JDBC, Typical uses of JDBC, The Structured Query language, Basic JDBC Programming concepts, Executing Queries.

1. Text Book :

   Java2 : The Complete Reference by Herbert Schildt, Tata McGraw- Hill, fourth Edition, 2005

Effective for Students Admitted in July, 2011 onwards

2. Reference books :
    1. JAVA How to Program by Deitel & Deitel, Pearson Education, Seventh
    2. Thinking in Java by Bruce Eckel , Prentice Hall, Third Edition, 2005
    3. Core Java 1.2: Volume 1 Fundamentals by Gary Cornell, Cay Horstmann, Prentice Hall, Seventh Edition, 2007

Class Assignments:

1. Write a program that produces the following output:
   Hello World!
   “It's been nice knowing you “.
   Goodbye world!
2. State the order of evaluation of the operations in each of the following Java statements and implement them to show the value of x after each statement.
   x = 7 + 3 * 6 / 2 – 1;
   x = 2 % 2 + 2 * 2 – 2 / 2;
   x = (3 * 9 * (3 + (9 * 3 / (3))));
3. Write an application that declares 5 integers, determines and prints the largest and smallest in the group.
4. Write an application that takes 3 parameters as sides of triangle and calculate area of triangle.
5. Write an application that declares two integers, determines whether the first is a multiple of the second and print the result. [Hint: Use the remainder operator.]
6. Write a program to find all prime numbers between 100 to 1000.
7. Write a program to check whether the given number is palindrome or not.
8. Write an application that evaluates the factorial of the integers from 1 to 5.
9. Write a program that accepts an integer from user and check whether the number is Fibonacci number or not.
10. Read a positive integer value, and compute the following sequence: If the number is even, halve it; if it's odd, multiply by 3 and add 1. Repeat this process until the value is 1, printing out each value. Finally print out how many of these operations you performed.
    Typical output might be:

```
Initial value is 9
Next value is 28
Next value is 14
Next value is 7
Next value is 22
Next value is 11
Next value is 34
Next value is 17
Next value is 52
Next value is 26
Next value is 13
Next value is 40
Next value is 20
Next value is 10
Next value is 5
Next value is 16
Next value is 8
Next value is 4
Next value is 2
Final value 1, number of steps 19
```

    If the input value is less than 1, print a message containing the word Error and perform an exit(0);

11. Write a program which will read an integer value for a base, then read a positive integer written to that base and print its value. Read the second integer a character at a time; skip over any leading nonvalid (i.e. not a digit between zero and ``base-1") characters, then read valid characters until an invalid one is encountered.

```
Input Output
========= ======
10 1234 1234
8 77 63 (the value of 77 in base 8, octal)
2 1111 15 (the value of 1111 in base 2, binary)
```

    The base will be less than or equal to 10.
12. write application that take a series of integers in pair(e.g. (a,b)) and find the following
    a. Is first one is greater
    b. Is both are even
    c. They are relatively prime or not
13. Write an application that uses *String* method *compareTo* to compare two strings defined by the user.
14. Write an application that uses String method *equals* and *equalsIgnoreCase* to tests any two string objects for equality.
15. Write an application that uses *String* method *indexOf* to determine the total number of occurrences of any given alphabet in a defined text.
16. Write an application that uses *String* method *concat* to concatenate two defined strings.
17. Write an application that finds the length of a given string.
18. Write an application that uses *String* method *charAt* to reverse the string.
19. Write an application that finds the substring from any given string using *substring* method and *startsWith* & *endsWith* methods.
20. Write an application that changes any given string with uppercase letters, displays it , changes it back to lowercase letters and displays it.
21. Create a class called *Employee* that includes three pieces of information as instance variables – a first name (type String), a last name (type String) and a monthly salary (double).
22. Create a constructor in above class to initialize the three instance variables. Provide a get method for each instance variable.
23. Write a test application named *EmployeeTest* that demonstrates class Employee's capabilities. Create two employee objects and display each object's yearly salary.
24. Give each employee a 10% raise and display each Employee's yearly salary again.
25. Create a class *Account* with an instance variable *balance* (double). It should contain a constructor that initializes the *balance*, ensure that the initial balance is greater than 0.0.
26. Create two methods namely *credit* and *getBalance*. The first one adds the amount (passed as parameter) to balance and does not return any data. The second method allows clients (i.e. the other classes that use this class) to obtain the value of a particular *Account* object's *balance*.
27. Create class *AccountTest* to create and manipulate an *Account* object.
28. Write another method *debit* in the above program that withdraws money from an Account. Ensure that the debit amount does not exceed the Account's *balance*. In that case the *balance* should be left unchanged and the method should print a message indicating "Debit amount exceeded account balance". Modify class *AccountTest* to test method debit.
29. Write Java program to create five object of book, get information of book using getdata() function including name, price, publication and author.
30. Write search() function to search a specified book, if book is search return the complete information of book and print the information of book using putdata() function.

31. Write an application to create a super class Employee with information first name & last name and methods getFirstName(), getLastName() derive the sub-classes ContractEmployee and RegularEmployee with the information about department, designation & method displayFullName() , getDepartment, getDesig() to print the salary and to set department name & designation of the corresponding sub-class objects respectively.
32. Derive sub-classes of ContractEmployee namely HourlyEmployee & WeeklyEmployee with information number of hours & wages per hour, number of weeks & wages per week respectively & method calculateWages() to calculate their monthly salary. Also override getDesig () method depending on the type of contract employee.
33. Write an application to create a super class Vehicle with information vehicle number, insurance number, color and methods getConsumption() and displayConsumption(). Derive the sub-classes TwoWheeler and FourWheeler with method maintenance() and average() to print the maintenance And average of vehicle.
34. Extend the above TwoWheeler class with methods getType() and getName() which gives the information about the type and the name of the company.Create sub-classes Geared and NonGeared with method average() to print the average of a geared and non-geared two wheeler.
35. Create a super class Shape with methods getName() which gives the information about the type of the shape.derive its sub-classes TwoDim and ThreeDim with method area() and volume() respectively which prints the area and volume of a two-dimensional and three-dimensional shape.
36. Extend the class TwoDim with methods getLength(),getBreadth() which displays the length and breadth of two dimentional shapes.Derive subclasses rectangle,rhombus with method getArea() and getPerimeter() to calculate the area and perimeter of this two dimensional shapes.
37. Extend the class ThreeDim with methods getLength(),getBreadth(),getHeight() which displays the length , breadth and height of three dimentional shapes.Derive sub-classes cuboid,tetrahedron with method getArea() and getVolume() to calculate the area and volume of this threee dimensional shapes.
38. Create a super class Student with methods getQual (), getFirstName(),getLastName(), getAddress(), getContat(), which gives basic details of student.derive sub-classes Faculty and Scholar with method salary(), Course() resp. which gives the additional information about the salary and course of faculty and scholar resp.
39. Create an abstract class Shape which calculate the area and volume of 2-d and 3-d shapes with methods getArea and getVolume. Reuse this class to calculate the area and volume of square ,circle ,cube and sphere.
40. Create an abstract class Employee with methods getAmount() which displays the amount paid to employee. Reuse this class to calculate the amount to be paid to WeeklyEmployeed and HourlyEmployee according to no. of hours and total hours for HourlyEmployee and no. of weeks and total weeks for WeeklyEmployee.
41. Create an Interface payable with method getAmount ().Calculate the amount to be paid to Invoice and Employee by implementing Interface.
42. Create an Interface Vehicle with method getColor(),getNumber(),getConsumption(). Calculate the fuel consumed, name and color for TwoWheeler and FourWheeler by implementing interface Vehicle.
43. Create an Interface Fare with method getAmount() to get the amount paid for fare of travelling. Calculate the fare paid by bus and train implementing interface Fare.
44. Create an Interface StudentFee with method getAmount(), getFirstName(),getLastName(), getAddress(), getContact(). Calculate the amount paid by the Hostler and NonHostler student by implementing interface StudentFee.

45. Write a Program to create your own package. Package should have more than two classes. Write a class that uses the package.
46. Exception Handling program for division of two numbers that accepts numbers from user.
47. Exception Handling program for storing values in array of *int* or *String* that results into buffer overflow.
48. Exception Handling program for calculating roots of quadratic equation that accepts coefficients from user.
49. Exception Handling program for *NullPointerException*--thrown if the JVM attempts to perform an operation on an *Object* that points to no data, or *null*
50. Exception Handling program for *NumberFormatException*--thrown if a program is attempting to convert a string to a numerical datatype, and the string contains inappropriate characters (i.e. 'z' or 'Q')
51. Exception Handling program for *ClassNotFoundException*--thrown if a program can not find a class it depends at runtime (i.e., the class's ".class" file cannot be found or was removed from the CLASSPATH).
52. Exception Handling program for *IOException--actually contained in java.io*, but it is thrown if the JVM failed to open an I/O stream.
53. Write a program that shows that the order of the catch blocks is important. If you try to catch a superclass exception type before a subclass type, the compiler should generate errors.
54. Program for demonstrating the use of throw, throws & finally - Create a class with a **main( )** that throws an object of class **Exception** inside a **try** block. Give the constructor for **Exception** a **String** argument. Catch the exception inside a **catch** clause and print the **String** argument. Add a **finally** clause and print a message to prove you were there.
55. Create your own exception class using the **extends** keyword. Write a constructor for this class that takes a **String** argument and stores it inside the object with a **String** reference. Write a method that prints out the stored **String**. Create a try-catch clause to exercise your new exception.
56. Write a program to rethrow an exception – Define methods one() & two(). Method two() should initially throw an exception. Method one() should call two(), catch the exception and rethrow it Call one() from main() and catch the rethrown exception.
57. Write a program to change the priority of thread.
58. WAP for producer consumer problem (w/o synchronization).
59. WAP for producer consumer problem (with synchronization).
60. Create an application of cash withdrawal from the bank account that have no. of users that are operating the accounts.(synchronization)
61. Open a text file so that you can read the file one line at a time. Read each line as a **String and** send the results to **System.out**.
62. Modify Exercise 1 so that the name of the file you read is provided as a command-line argument.
63. Modify Exercise 2 to force all the lines in the results to upper case and send the results to **System.out.**
64. Modify Exercise 2 to also open a text file so you can write text into it.
65. Implement a pair of classes, one Reader and one Writer, that count the number of times a particular character, such as e, is read or written. The character can be specified when the stream is created.
66. Construct a program Wc ("word count"), which counts number of chars, words and lines of the text file. Space is counted as a character. Empty rows are counted as lines. "Word" will represent a string.
67. Create an application to display a frame with title MyFrame.
68. Create an application to draw a horizontal line.

69. Create an application to draw one line perpendicular to other. One line parallel to other.
70. Create an application to display a circle within rectangle.
71. In the above application fill different colors in the circle & rectangle.
72. Write an application that displays any string. Choose color from combo box to change the color of this displayed string and choose its size & type respectively from another two combo boxes.
73. Write a small application with a default date 01/01/2000 and three combo boxes displaying valid days, months & year (1990 – 2050). Change the displayed date with the one chosen by user from these combo boxes.
74. Create a GUI with a text field and three buttons. When you press each button, make some different text appear in the text field.
75. Create a GUI application to take input of two numbers(text field) from user. When you press button it should display sum of the two numbers in a third text box.
76. Create an applet with a Button and a TextField. Write a referenceEvent( ) so that if the button has the focus, characters typed into it will appear in the TextField.
77. Write an application to create a GUI with two buttons such that clicking on the first displays the message "Welcome to SCS" on the window and clicking on the second changes the color of the message(*hint : toggle the color*)
78. Create a GUI with title STUDENT which has labels roll no., name, class, address with textboxes for taking input from the user(without any functionality).
79. Create a GUI application for fees receipt which contains checkboxes for selecting the course, radio buttons for selecting gender and labels and corresponding textboxes for name, class, date and amount paid.
80. Create a GUI application to display a calculator using grid Layout (You do not have to provide functionality).
81. WAP for string tokenizer.
82. Create a program that will print every other argument given on the command line. (Use of string tokenizer), consider how your program will deal with no argument.
83. WAP that generate a random number (1 – 10000). Let the user guess the correct number. User will enter the digit. Program should let the user that input is right or wrong. No of turns that user can make a choice for input is twice the number of digits in the system generated numbers.Convert the input date in words. Input format is dd mm yy.
84. Find the frequency of each number in the array.

CS - 302 Database Applications and Tools

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Database Environment:** Data versus information, traditional file processing, disadvantages, database approach, range of database application, advantages of database approach. Cost and risk factors, components of database environment, evolution of database system.
Unit 2	**Database Development Process:** Information engineering, information architecture, enterprise data model, planning, SDLC, CASE etc. Steps of planning, strategic planning factors, corporate planning objects. Developing preliminary data model, and use of planning matrices, SDLC steps, CASE role, people in database development, three-schema architecture for database development. Examples to demonstrate the development process. **Modeling Data in the Organisation:** Modeling of the rules of organization, data names and definitions, ER model constructs entities and its types, attributes, relationships, degree, unary, binary, ternary, n-ary, cardinalities constraints, ER modeling examples.
Unit 3	**Enhanced ER modeling:** supertype, subtypes, specialization, generalization, specifying constraints in EER models, completeness, Disjointness, discriminators, defining super/sub type hierarchies, EER modeling examples, live demos modeling for few scenarios. Logical database design and relational model development, Relational model properties, keys, primary, secondary, composite, properties of relations. Codd's rules, integrity constraints, creating relational tables, Transform EER diagrams into relations, seven different steps for mapping EER model into relations,
Unit 4	Introduction to normalization, steps, functional dependencies, basic normal forms, definition of first, second, third normal form and removing anomalies from the relations. De-normalization and merging relations.
Unit 5	**Special Topics (Overview) :** Data Warehousing, Data Mining, Distributed Databases, Object oriented modeling, definitions, activities in phases of model development, advantages of OOM, UML class diagrams, Example of a model development.

1. Text Book :
   "Modern Database Management" Seventh Edition, Hoffer, Prescott, McFadden Pearson Education"
2. Reference books :
   1. Database Systems "Thomas M. Connolly, Carolyn E. BeggPearson Education.
   2. Raghu R and Johannes G., "Database management Systems", Mc Hill 3rd Ed 2002, Elmasri R, Navathe S, "Fundamentals of Database Systems", Addison Wesley 4th Ed.

Class Assignments:

Exercise 1: Make a team of three or four. Choose any of the projects and submit a well documented short description. Also give the plan for making the project successful.

Exercise-2: For any project of your choice, develop the data development approach and prepare a report for the same, presentation will be necessary.

Exercise-3: For the project of your choice, describe the development of ER/EER model and document the complete conceptual design along with the presentation.

Exercise-4: For the project of your choice, describe the development of normalized relational model from the ER/EER model and document the complete relational design

Exercise-5: For the project of your choice, describe the SQL commands for creating all the tables, and sort and search of I/O information, using the knowledge gained from this chapter and the previous design.

Exercise 6: Full project report preparation and presentation of the project.

CS - 303 Theory of Computation

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**The Theory Of Automata:** String, Alphabets and Languages, Finite Automata, Finite State Machine, Basic Definition. Description of a Finite Automaton, Deterministic Finite Accepters- Transition Graphs, Languages, Non- Deterministic Finite Acceptors- Definition, Finite Automata with ϵ- moves Equivalence of Deterministic and Nondeterministic Finite Accepters, Mealy and Moore models-Definitions, Transformation of Mealy Machine into Moore Machine and vice-versa.
Unit 2	Conversion of NDFA to DFA Removal of ϵ transition from ϵ – NDFA. The Myhill-Nerode theorem and Minimization of Finite Automata – Definition and Construction. **Properties of Regular Sets:** Pumping lemma for regular set, Closure properties of regular set.
Unit 3	**Formal Language:** Basic Definition, Chomsky Classification of languages, Initialization of Finite Automata Regular Expression and Language Regular Expressions, Connection between Regular Expressions and Regular Languages Regular Grammars – Right and Left Linear Grammars, Equivalence between Regular Languages and Regular Grammars
Unit 4	**Context-Free Grammars:** Leftmost and Rightmost Derivations, Derivation Trees, Parsing and Ambiguity, Simplification of CFGs, Chomsky Normal Form, Greibach Normal Form, Cocke-Kasami-Younger Algorithm, Properties of Context-Free Languages.
Unit 5	**Pushdown Automata:** Definition, Non deterministic Pushdown Automata, Pushdown Automata for Context Free Languages. Context-Free Grammars for Pushdown Automata. Deterministic Pushdown Automata and Deterministic Context-Free Languages. **Turing Machine:** Definition of Standard Turing Machine, Turing Machine as Language Accepters and Transducers.

1. Text Book :
   Mishra and Chandrasekaran, Theory of Computer Science (Automata, language and Computation), 2nd Ed. Prentice Hall of India.
2. Reference books :
   1. Martin, J.C.: Introduction to Languages and the Theory of Computation, McGraw- Hill, Inc., 3rd ed., 2002. ISBN 0-072-32200-4.
   2. Brookshear, J.G.: Theory of Computation: Formal Languages, Automata, and Complexity, The Benjamin/Cummings Publishing Company, Inc, Redwood City, California, 1989. ISBN 0-805-30143-7
   3. Peter Linz, An Introduction to Formal Languages and Automata, Narosa Publishing House.

Class Assignments:

1. Write a C program to generate strings up to length 15 with the indivisible symbols like {0,1}, {a, b}, {+,-}.
2. Write a C program that accepts any Context Free Grammar (use appropriate data structure to store CFG) and then print it in the proper format.
3. Write a program for recognition of a given string using CKY algorithm. Use a subprogram for generation of string (*) of various length. (Length u to 8 and * of 0,1).
4. Write a C/C++ program which reads a program written in any programming language (say C/C++/Java) and then perform lexical analysis. The output of program should contain the tokens i.e. classification as identifier, special symbol, delimiter, operator, keyword or string. It should also display the number of identifiers, special symbol, delimiter, operator, keyword, strings and statements.

CS - 304 Computer Graphics & Multimedia

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Introduction to Computer Graphics**, Application of Graphics, Display Devices: Refresh Cathode -Ray Tubes, Raster Scan Displays, Random Scan Displays, Color CRT Monitors, Flat Panel Displays. Video cards/display cards Input Devices: Mouse, Trackball, Space ball, Data Glove, Joystick, Light pen, Scanner, Digital Camera, Touch Panels, Voice Systems.Hardcopy Devices: Printers and Plotters
Unit 2	**Graphics Primitives:** Line Generation Algorithms: DDA algorithm, Bresenham's algorithm. **Graphics Primitives:** Circle Generation Algorithms: Midpoint Circle algorithm, Bresenham's circle generation algorithm. Ellipse Generation algorithm. **Graphics Primitives:** Polygon filling Algorithms: Scan Line Polygon fill algorithm, Inside - Outside Tests, Boundary-Fill algorithm, Flood - Fill algorithm. Fundamentals of aliasing and Antialiasing Technique.
Unit 3	**Clipping:** Clipping operations, Point clipping, Line clipping: Cohen Sutherland Algorithm, Liang Barsky Algorithm, Nicholl-Lee-Nicholl Algorithm. **Polygon clipping:** Sutherland- Hodgeman Algorithm, Weiler Atherton Algorithm. **Text clipping, Exterior clipping.**
Unit 4	**Two Dimensional Transformations:** Translation, Scaling, Rotation, Reflection, Shear, Homogenous coordinate system, composite transformations, raster method of transformation **Two Dimensional Viewing:** Window to Viewport coordinate transformation, **Three Dimensional:** 3D Geometry, 3D display techniques, transformations. **Projections:** Parallel Projection, Perspective Projection.
Unit 5	**Color Models and Color Application:** Color models: Properties of Light. Standard Primaries and the Chromaticity Diagram, XYZ Color Model, CIE Chromaticity Diagram. RGB Color Model, YIQ Color Model, CMY Color Model, HSV Color Model. Conversion between HSV and RGB Models. HLS Color Model, Color Selection and Application. **Advancements in the technology in Computer Graphics.**

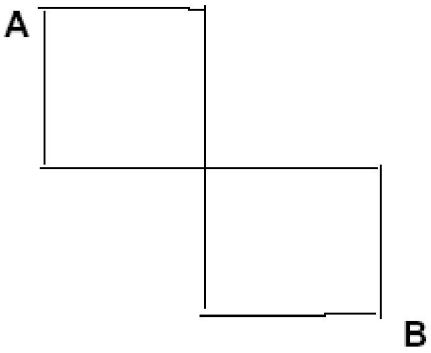
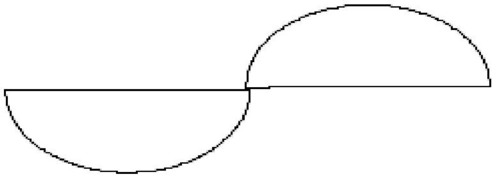
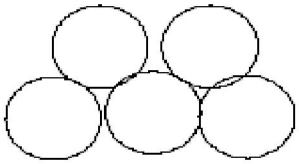
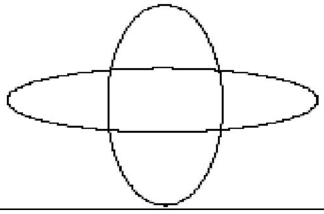
1. **Computer Graphics:** Donald Hearn and M. Pauline Baker, Second Edition, Prentice Hall of India.

Class Assignments:

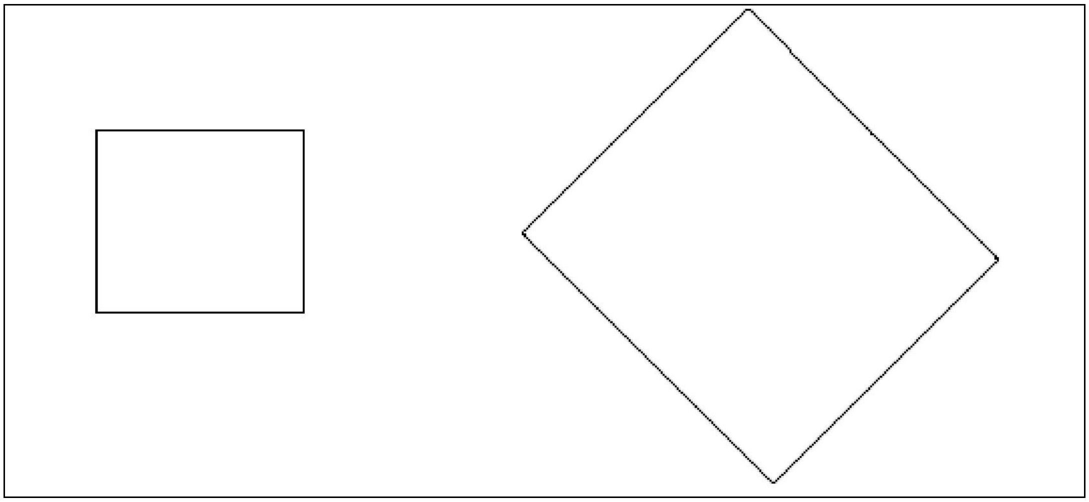
1. Write a program to implement DDA algorithm
2. Write a program to implement bresenham algorithm
3. Write a program to implement circle generating algorithm
4. Write a program to implement ellipse generating algorithm
5. Write a program to implement scaling on polygon.
6. Write a program to implement transferring on polygon.
7. Write a program to implement rotation on polygon.
8. Write a program to implement reflection on polygon.
9. Write a Program to implement set of Basic Transformations on Polygon i.e. Translation, Rotation and Scaling.
10. . Write a program to implement set of Composite Transformations on Polygon i.e Reflection, Shear (x &Y), rotation about an arbitrary point.
11. Write a program to implement Line Clipping Algorithm using Cohen Sutherland Algorithm.
12. Write a program to implement Line Clipping Algorithm using Liang Barsky Algorithm.
13. Write a program to Implement Polygon Clipping Algorithm using sutherland Hodgman Algorithm

Effective for Students Admitted in July, 2011 onwards

14. What are the advantages of Bresenhams algorithm over DDA algorithm?
15. How can the Bresenham's algorithm be modified to accommodate all types of lines?
16. Modify the BRESENHAM algorithm so that it will produce a dashed-line pattern. Dash length should be independent of slope
17. What is the significance of homogeneous co-ordinates? Give the homogeneous coordinates for the basic transformations.
18. Why are matrices used for implementing transformations?
19. Show that two successive reflections about any line passing through the coordinate origin is equivalent to single rotation about the origin.
20. Determine the sequence of basic transformations that are equivalent to the x-direction and y-direction shearing matrix.
21. Show that transformation matrix for a reflection about the line y=x, is equivalent to a reflection relative to the x axis followed by a counterclockwise rotation of 90 degrees.
22. 22. Modify the Liang-Barsky line clipping algorithm to polygon clipping.
23. 23. Write a routine to clip an ellipse against a rectangular window.
24. 24. Write a routine to implement exterior clipping on any part of a defined picture using any specified window.
25. Distinguish between viewport and window.
26. Write a Short notes on viewing transformation.
27. What do you mean by normalization transformation? Why it is needed?
28. Explain the Sutherland and cohen subdivision algorithm for the line clipping.
29. Problems on Sutherland and cohen subdivision algorithm for the line clipping.
30. Explain Liang-Barsky line clipping algorithm
31. Problems on Liang-Barsky Line clipping algorithm.
32. Explain Sutherland â€“ Hodgeman algorithm for polygon clipping.
33. What do you mean by interior and exterior clipping?
34. Explain how exterior clipping is useful in multiple window environment.
35. Derive the transformation matrix for rotation about an arbitrary axis.
36. Derive the transformation matrix for rotation about an arbitrary plane
37. What are different types of projection? Derive a matrix representation for perspective transformation? What are different perspective anomalies?
38. Problems on 3-Dimension transformation. For eg A triangle is defined by 3 vertices A,B,C .Find the final co-ordinates after it is rotated by some degree around line joining the points(coordinates will be given).
39. What are the characteristics of Video Display Devices?
40. Compare and contrast the operating characteristics of Raster Refresh Systems, Plasma Panels and LCDs.
41. Compare Refresh type and Storage type CRT display.
42. Explain different types of kernel systems
43. Application of CG in Education and Training
44. Write detailed note on frame buffer organization.
45. Write short note on different input devices.
46. What are the advantages of Bresenhams algorithm over DDA algorithm.
47. Modify the BRESENHAM algorithm so that it will produce a dashed-line pattern. Dash length should be independent of slope.
48. Write a procedure to scan the interior of a specified ellipse into a solid color.
49. Modify the 4-connected boundary fill algorithm to avoid excess stacking.
50. Write the Scan line filling algorithm
51. Write a program to draw the following figure:-

fig 1:- All sides are equal, points A and B are the inputs.  	fig:-2 Circles with radius "r"  fig:-3 Circles with radius "r"  fig:-4 ellipse semimajor axis = rx and semn 

52. What is the significance of homogeneous co-ordinates? Give the
53. homogeneous co-ordinates fot the basic transformations.
54. Why are matrices used for implementing transformations.
55. Show that two successive reflections about any line passing through the coordinate origin is equivalent to single rotation about the origin.
56. Determine the sequence of basic transformations that are equivalent to the x-direction and y-direction shearing matrix.
57. Show that transformation matrix for a reflection about the line y=x, is equivalent to a reflection relative to the x axis followed by a counterclockwise rotation of 90 degrees.
58. Find a transformation of triangle (coordinates will be given) by Rotating
59. 45^0 degree about the origin and then translating one unit in X and Y direction
60. Examples on Rotation with respect to arbitrary point.
61. Derived transformation matrix for the following figure.



Effective for Students Admitted in July, 2011 onwards

CS - 305 Computer Networks

Max. Marks : 50 **Min. Marks : 17**

Unit	TOPIC
Unit-1	Introduction: Computer Network, Goals and Applications, Reference models – OSI and TCP/IP. A Comparative study. LAN, MAN and WAN and topologies, LAN components – File server, Workstations, Network Adapter Cards. Connection Oriented and Connection less services, Switching Techniques – Circuit Switching, Packet Switching.
Unit-2	Data Link Layer**:** Design Issues, Framing, Error Detection: Parity Check, Check Sum and Cyclic Redundancy Check (CRC); Correction Technique: Hamming code. Flow Control: Elementary Data Link Protocols: An Unrestricted Simplex Protocol, Simplex Stop-and-Wait Protocol, Sliding Window Protocols: One-Bit Sliding Window Protocol Go Back N and Selective Repeat. Data link layer in the Internet: SLIP and PPP.
Unit-3	MAC Sublayer**:** Multiple access protocols: Aloha, CSMA Protocols; Collision-Free Protocols; IEEE MAC Sublayer protocols: 802.3 802.4, 802.5 and their management. High speed LANs – Fast Ethernet, FDDI, Wireless LANs
Unit-4	Network Layer: Design issues, Routing Principles. Routing Algorithms: Optimality Principle, Shortest Path Routing, Flooding, Distance Vector Routing. Link State Routing, Hierarchical Routing, Broadcasting Routing, Multicast Routing, The Network Layer in the Internet: Internet Protocol, Internet addressing and Internet Control protocols.
Unit-5	Transport Layer: Elements of Transport Protocols, Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing. The Internet Transport Protocol UDP: Introduction to UDP. Introduction to TCP, The TCP Service Model, The TCP Protocol. Application layer**:** Client Server Architecture, DNS, WWW and HTTP, Cookies, Proxy Server. E-mail Protocols (SMTP, POP3, IMAP, MIME), FTP, TELNET. Network Security: Cryptography, Symmetric- key Algorithms, Public- key Algorithms, Digital Signatures.

1. Required Text
   - Computer Networks, Andrew S. Tanenbaum, Addison-Wesley, 4th Ed.

2. Essential References
   - Data Communications and Networking, B.A. Frouzan, McGraw-Hill.

Class Assignments:

Assignment I

1. Define the following

i) Computer networks	ii) Autonomous computer
iii) Distributed system Vs Computer networks	iv) LAN, MAN, WAN
vi) Protocols, interfaces and services	v) Internet
vii) Host or end system	viii) Subnet

2. Write the need of layered architecture.
3. Explain OSI and TCP/IP Model. Also write the development stages of these models(i.e. When these model came., who proposed and further development).
4. Compare OSI and TCP/IP Model.

Assignment II

1. A bit string, 01111011111011111110, needs to be transmitted at the data link layer. What is the string actually transmitted after bit stuffing?
2. An 8-bit byte with binary value 10101111 is to be encoded using an even-parity Hamming code. What is the binary value after encoding?
3. What is the remainder obtained by dividing $x^7 + x^5 + 1$ by the generator polynomial $x^3 + 1$?
4. What is the mechanism of sliding window flow control?

Assignment III

1. Consider the delay of pure ALOHA versus slotted ALOHA at low load. Which one is less? Explain your answer.
2. Sketch the Manchester encoding for the bit stream: 0001110101.
3. Describes Carrier Sense Multiple Access Protocols in short?
4. Difference between slotted and pure aloha?

Assignment IV

1. Give two example computer applications for which connection-oriented service is appropriate. Now give two examples for which connectionless service is best.
2. Describes distance vector and link state routing algorithm?
3. What are the functions of router, bridges, repeaters, and gateways?
4. Difference between adaptive and non adaptive routing?
5. Describes Dijkstra algorithm?

Assignment V

1. Explain different classes of in Internet address?
2. Explain purpose of ARP, RARP, ICMP, and IGMP?
3. What is host id and net id?
4. What is the purpose of sub netting?
5. Explain IP datagram header format?
6. Explain UDP and TCP datagram header format?

Assignment VI

1. Write in short about following:-
   - TELNET
   - FTP
   - SMTP
   - POP
   - MIME
   - HTTP
   - URL
   - CGI
   - DNS

Effective for Students Admitted in July, 2011 onwards

CS - 401 Linux/ Unix Administration

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Background:** Evolution of Unix OS. Unix implementations. Features of Unix operating system. **Linux operating system:** Development of Linux. Applications of Linux operating system.
Unit 2	**Basic UNIX environment**: Basic commands, directory management, pipes, tee, I/O redirection and other utilities. **Advanced commands:** File system and process management commands, Shell, Pattern matching, Navigating the File Systems.
Unit 3	**Unix editor:** VI editor, Creating new files. Text addition, deletion and changes. Dealing with sentences and paragraphs. Searching. Cut, paste and copy. Running C/C++ programs. **Shell programming**: Features of shell. Shell variables. Control statements. **Advance shell programming**: Command line arguments. Interactive shell scripts. Debugging of shell scripts. Communication facilities in Unix.
Unit 4	**Structure of unix operating system**: Structure of unix kernel, Unix system calls. **Unix system**: File system calls, Process management calls. **Advance Filter:** Awk: Number processing, Interface with shell, functions.
Unit 5	**Unix system administration:** Adding and removing users. User accounting. Adding and removing hardware. Performing backups and restore. Disk space management. **Unix system administration**: Configuring the kernel. Network management in Unix. Performance analysis. Unix Desktop.

1. Text Book :

   UNIX Operating Systems: Sumitabh Das, Tata McGraw Hills publication.
2. Reference books :
   1. **UNIX System Administration Handbook**(Second edition): Evi Nemeth, Garth Synder, Scott Seebass, Trent R Hein, Pearson Education - Asia, 2000.
   2. C: **Design of UNIX Operating System**: Maurice J. Back, Pearson Education - Asia.

LAB Assignments:

Aim: The aim of this laboratory is to get students acquainted with Unix/Linux commands, shell programming & system administration.

LAB -1 Information management

Cal, date, tty, sh, env, set man ,who, whoami

Basic Operations

- i. Connecting to the system
- ii. Disconnecting from the system
- iii. Text and graphic mode
- iv. Changing your password
- v. Navigating through the file system
- vi. Determining file type
- vii. Looking at text files
- viii. Finding help
- ix. List the different types of file comparisons command.

LAB- 2 Utility commands

Wc,eco,tail,less,more,short,grep,bc,cmp,comm.

LAB-3 File system management
Ls,ln,rn,rmdir,mkdir,file,chmod,find,od,pwd,locate,updated,mount,umount,mv

LAB- 4 Process management
Ps,kill

LAB- 5 Compilation and debugging
cc, gdb

LAB- 6 Editor
Vi,joe,mcedit,emac

LAB -7 Using unix commands –

a) Count the number of users who logged in and display the result.
b) Count the number of files in the current directory
c) Count the number of differences(in term of lines) between two files

Shell programming

LAB- 8

1. Execute the following shell variables
   $HOME
   $PATH
2. concatenate two shell variables

LAB -9

1. Write a shell script to find the sum of first N natural numbers
2. Write a shell script to read n numbers from the user and display its sum.
3. Write a shell script to read 10 numbers from the user and to find the sum and Average of the numbers.

LAB-10

1. Write a shell script to generate even numbers and to calculate its sum .
2. Write a shell script for swapping two numbers (using only two variables) .
3. Write a shell script to find the count and sum of even and odd numbers separately.

LAB-11

1. Write a shell script to find the factorial of a given number.
2. Write a shell script to find the sum of digits in an integer .
3. Write a shell script to count the number of digits in an integer .

LAB-12

1. Write a shell script to find whether the given number is Armstrong or not .
2. Write a shell script to print the multiplication table of the given number .
3. Write a shell script to print all divisors of a given integer .

LAB-13

1. Write a shell script to generate Fibonacci series.
2. Write a shell script to find the length of the string.
3. Write a shell script to extract a sub string from a given string.

CS - 402 Compiler Design

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	Compiler, Translator, Interpreter definition, Phase of compiler introduction to one pass & Multi pass compilers. Analysis of source program, duties of 6 phases. Review of Finite automata lexical analyser, Input, buffering, Recognition of tokens, Idea about LEX: A lexical analyser generator, Error handling.
Unit 2	Introduction to parsing. Bottom up parsing Top down parsing techniques. Shift reduce parsing, Operator precedence parsing, Recursive descent parsing predictive parsers. LL grammars & passers error handling of LL parser. LR parsers, Construction of SLR.
Unit 3	Conical LR & LALR parsing tables, parsing with ambiguous grammar, Syntax directed definitions; Construction of syntax trees, L~attributed definitions, Top down translation. Intermediate code forms using postfix notation and three address code.
Unit 4	Representing TAC using triples and quadruples, Translation of assignment statement. Boolean expression and control structures. Definition of basic block control flow graphs, DAG representation of basic block.
Unit 5	Advantages of DAG, Sources of optimization, Loop optimization, Idea about global data flow analysis, Loop invariant computation, Peephole optimization Issues in design of code generator, A simple code generator, Code generation from DAG. Code Optimization.

1. Text Book :
   Aho-Ullman, Principles of compiler Design, Narosa Publishing House.
2. Reference books :
   1. Aho-Ullman, Compilers: Principles Techniques & Tools, Addison Wesley.’
   2. Dhamdhere, Compiler Construction.

Class Assignments:

- Develop a lexical analyser to recognise a few patterns in C and C++.
  (ex: identifiers, constants, comments, operators etc.)
- Write a programme to parse using Brute force technique of Top down parsing.
- Develop on LL (1) parser (Construct parse table also).
- Develop an operator precedence parser (Construct parse table also)
- Develop a recursive descent parser

CS - 403 Internet and Web Technology

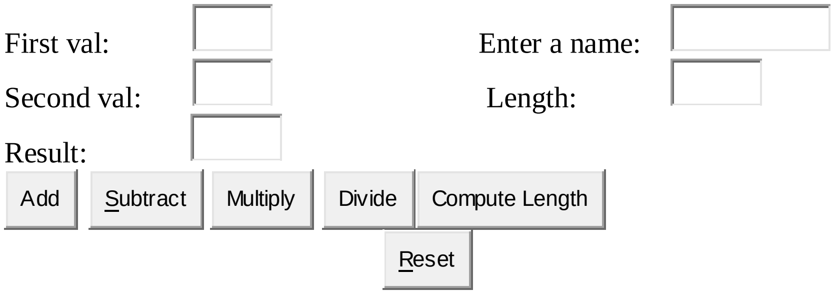
Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	Introduction Dynamic Web Programming, HTML Forms, scripting languages, Introduction to HTTP, web Server and application Servers, Installation of Application servers, Configuration files, Web.xml. Java Servlet, Servlet Development Process, Deployment Descriptors, The Generic Servlet Lifecycle.
Unit 2	Servlet Packages, Classes, Interfaces, and Methods, Handling Forms with Servlets. Various methods of Session Handling. Various elements of deployment descriptors. Java Database Connectivity: various steps in process of connection to the database, Various type of JDBC Driver.
Unit 3	Connection of JSP and Servlet with different database viz. Oracle, MS-SQL Server, MySQL. java.sql Package. Accessing metadata from the database. Type of Statement, Connection pooling: multiple users and need of connection pooling.
Unit 4	JSP Basics: JSP lifecycle, Directives, scripting elements, standard actions, implicit objects. Writing JSPs. Expression Language (EL), Separating Business Logic and Presentation Logic, Building and using JavaBean.
Unit 5	Session handling in JSP, Types of errors and exceptions handling, Standard Tab Library in JSP, Building Custom Tag Library, JSP Tag Library, MVC Design pattern Advances in J2EE and Other Web technology

1. Text Book :
   Kevin Mukhar, Chris Zelenak, James L Weaver, “Beginning Java EE 5: From Novice to Professional” Apress
2. Reference books :
   1. Marty Hall, Larry Brown, “Core Servlets and Java Server Pages”, 2nd edition, Pearson Education
   2. JavaDoc for various technologies
   3. Internet and Web technologies, TMH, 2002

Class Assignments:

1. Write a servlet that prints "Hello World"
2. Write a servlet that knows to whom it's saying hello, This servlet must be called from an HTML page taking user name as input. (Use both get and post method)
3. Write a servlet that counts and displays the number of times it has been accessed since the last server reboot.
4. Write a servlet that counts the times it has been accessed, the number of instances created by the server, and the total times all of them have been accessed.
5. Write a servlet that counts and displays the number of times it has been accessed, and reads an init parameter to know what at what number to begin counting.
6. This servlet counts and displays the number of times it has been accessed, and saves the count to a file in its destroy() method to make the count persistent.
7. Write a servlet that searches for prime numbers above one quadrillion. The algorithm it uses couldn't be simpler: it selects odd-numbered candidates and attempts to divide them by every odd integer between 3 and their square root. If none of the integers evenly divides the candidate, it is declared prime. It's disabled to let the server's CPU handle important tasks.
8. Write a servlet that prints the name and value for all of its init parameters.
9. Write a servlet that displays information about its server (The process is called Snooping).
10. Write a servlet that snoops the server's servlet and Java version.
11. Write a servlet to compute arithmetic operations on numbers and strings as follows:



First val:

Second val:

Result:

Enter a name:

Length:

12. Write a servlet that checks the client machine and only allows access if the client appears to be coming from somewhere other than the Terrorist 7 countries. A servlet like this could help restrict the export of strong encryption.
13. Write a servlet that prints its query string and then prints the name and value for all its parameters.
14. Write a servlet that prints the extra path information it receives and the resulting translation to a real path.
15. Write a servlet that serves files by using the getPathTranslated() and getMimeType() methods to return whatever file is given by the extra path information.
16. Write a servlet that prints information about its HTTP request headers.
17. Write a servlet that performs a random redirect, sending a client to a random site selected from its site list. Depending on the site list, a servlet like this could have many uses. As it stands now, it's just a jump-off point to a selectino of cool servlet sites. With a site list containing advertising images, it can be used to select the next ad banner.
18. Redirections can be used to learn where clients go when they leave your site. Assume you have several pages containing lists of links to other sites. Instead of linking directly to the external site, you can link to a redirecting servlet that can record each time an external link is selected. The HTML looks like this:
    <a href="/goto/http://www.servlets.com">Servlets.com</a>
    Write a servlet that can be registered to handle the /goto/* path prefix where it will receive the selected URL as extra path info and redirect the client to that location after making a note in the server log. servlet shows a servlet that uses client pull to display the current time, updated every 10 seconds.
19. Write a servlet that shows a servlet that uses client pull to display the current time, updated every 10 seconds.
20. Write a servlet that redirects requests for one host to another host, giving an explanation to the client before the redirection.
21. Write a servlet that demonstrates session tracking using hidden form fields by displaying the shopping cart for a booksite.
22. Write a servlet that uses session tracking to count the number of times a client has accessed it.
23. Write a servlet that demonstrates how to programmatically alter the current timeout. On first execution, the current timeout displays the application-wide setting. On second execution, the current timeout displays two hours—because that’s the timeout set during the first execution.
24. Write a servlet that snoops all the information about the current session.
25. Write a servlet that is protected by BASIC authentication as shown in web.xml and tomcat-users.xml. To see the salary information you'll need to login as a "manager" using names and passwords in tomcat-users.xml.
26. Write a servlet that lets a user vote for his favorite food from a combo box or radio buttons (the user must be able to make multiple food selections per request).
    - Store the favorite foods and the number of votes for each food.
    - Display all foods and their number of votes in alphabetical order back to the user.

    Use an appropriate Collection class or Map class to store the data.
27. Write a servlet that establishes a database connection using the values stored within the its sql.properties file having the following contents.

```
connection.driver=sun.jdbc.odbc.JdbcOdbcDriver
  connection.url=jdbc:odbc:somedb
   user=user
```

password=passwd

28. Write a servlet that uses the Oracle JDBC driver to perform a simple query, printing names and phone numbers for all employees listed in a database table. Here assume that the database contains a table named EMPLOYEES, with at least two fields, NAME and PHONE.

29. Write a JSP to output the values returned by System.getProperty for various system properties such as java.version, java.home, os.name, user.name, user.home, user.dir

30. Write a JSP to output the entire line, "Hello! The time is now ..." but use a scriptlet for the complete string, including the HTML tags.

31. Write a JSP to output all the values returned by System.getProperties with "<BR>" embedded after each property name and value. Do not output the "<BR>" using the "out" variable.

32. Modify exercise # 28-30 to import the java.util packages.

33. Write a JSP to do either a forward or an include, depending upon a boolean variable.

34. Write a JSP/HTML set that allows a user to enter the name of a system property, and then displays the value returned by System.getProperty for that property name (handle errors appropriately.)

35. Make a JSP page that randomly selects a background color for each request. Just choose at random among a small set of predefined colors. Be sure you do not use the JSP-Styles.css style sheet, since it overrides the colors given by
<BODY BGCOLOR="...">.

36. Make a JSP page that lets the user supply a request parameter indicating the back-ground color. If no parameter is supplied, a background color should be selected at random.

37. Make a JSP page that lets the user supply a request parameter indicating the back-ground color. If no parameter is supplied, the most recently used background color (from a previous request by any user) should be used.

38. The java.math package has a class called BigInteger that lets you create whole numbers with an arbitrary number of digits. Create a JSP page that makes a large BigInteger from a String you supply as a request parameter, squares it, and prints out the result. Use the online API at http://java.sun.com/j2se/1.5.0/docs/api/ to see the syntaxfor the BigInteger constructor and squaring operations.

39. Make an HTML “signature” block with your name and email address. Include it in two JSP pages.

40. The value of the page attribute of jsp:include is allowed to be a JSP expression. Use this idea to make a JSP page that includes a “good news” page or a “bad news” message at random.

41. Suppose that you have two different JSP pages that do two different things. However, for both pages you want to let the user supply a bgColor attribute to set the background color of the page. Implement this, but use an include mechanism to avoid repeating code. For example:

    White background: http://host/path/page1.jsp

    White background: http://host/path/page2.jsp

    Red background: http://host/path/page1.jsp?bgColor=RED

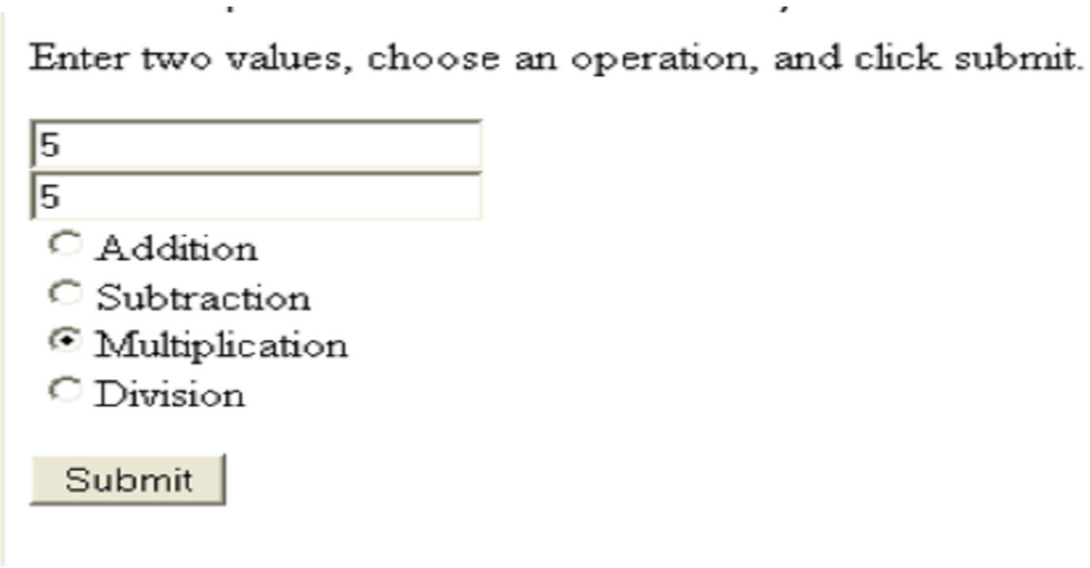
42. Yellow background: http://host/path/page2.jsp?bgColor=YELLOW

43. Make two separate JSP pages that have bulleted lists containing random integers in a certain range. Avoid repeating code unnecessarily by including a page that defines a randomInt method.

44. Define a class called ColorBean that stores strings representing a foreground color and a background color. Compile and test it separately (i.e., without using a servlet or JSP page). Note: if your tester class (i.e., the one that has “public static void main(String[] args) {...}” in it) is in a package, remember that you have to use the package name when you run it from the command line. That is, you have to do “javac BeanTester.java” and then “java yourPackage.BeanTester”.

45. Make a “color preference” form that collects the user’s preferred foreground and background colors. Send the data to a JSP page that displays some message using those colors. This JSP page should use a default value for any form value that the user fails to supply (but don’t worry about empty strings). So, for example, if the user goes directly to the JSP page (bypassing the form), the JSP page should still work fine. For now, don’t worry about the user sending you whitespace; just handle totally missing values.

46. Redo the color preference example, but if the user fails to supply either of the colors, use whatever value they gave last time. If you have no previous value, use a default. (Hint: this problem is almost exactly the same difficulty as the previous one.)
47. Redo the color preference example, but if the user fails to supply any of the parameters, use whatever color the most recent user gave last time. Why could this give bad results?
48. Write a JSP that takes the user’s name and age from a form.
    - Echo back the name and age along with a message stating the price of movie tickets.
    - The price is determined by the age passed to the JSP.
    - If the age is greater than 62, the movie ticket price is $7.00.
    - If the user is less than 10 years old, the price is $5.00.
    - For everyone else, the price is $9.50.
49. Write a JSP that will allow a user to enter two values, select a type of mathematical operation to apply against them, and then upon clicking Submit, will display the result of the operation. An example of the default entry form:
50.



Enter two values, choose an operation, and click submit.

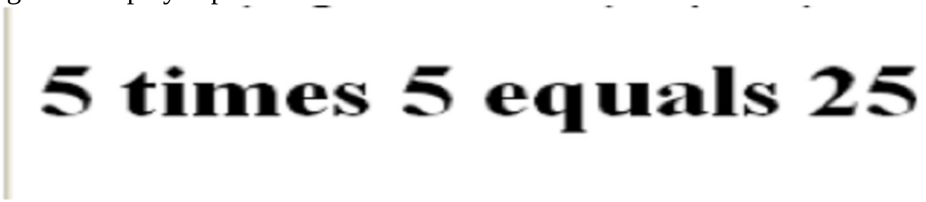
5

5

○ Addition
○ Subtraction
● Multiplication
○ Division

Submit

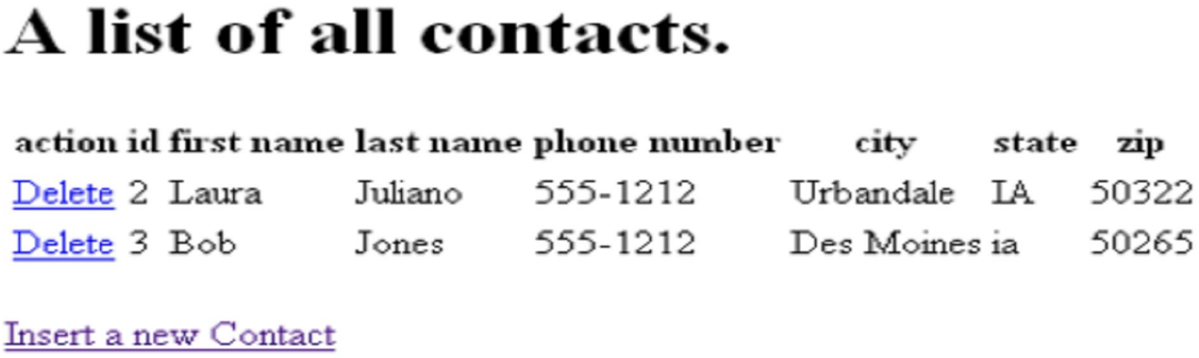
And the page that displays upon Submit:



5 times 5 equals 25

A valid code must also have:
- one jsp*only* - Math.jsp - that submits to itself
- four types of operations - add, subtract, multiply, divide
- display some kind of error if the user attempts to divide by 0

51. Create a database table to store contact information, then write a web application to manage viewing, adding, and deleting contacts from that table. The view should look similar to:



A list of all contacts.

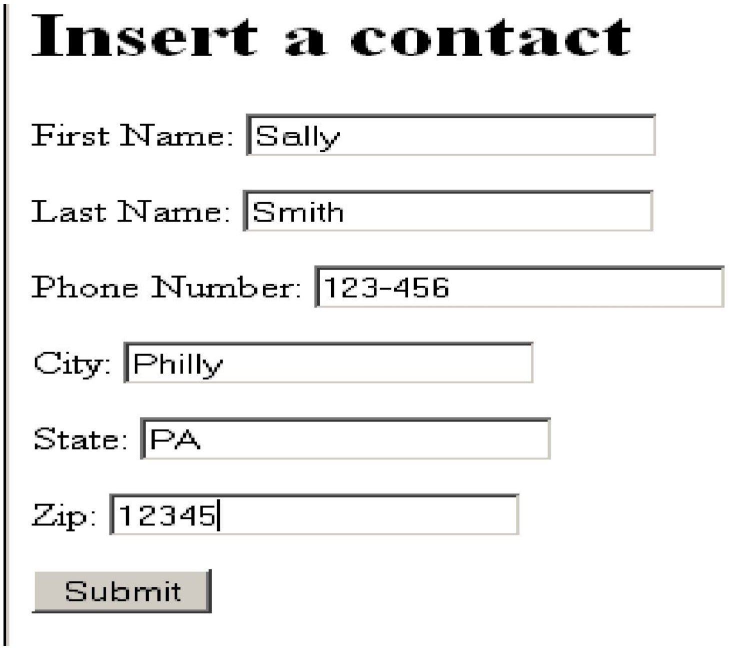
action	id	first name	last name	phone number	city	state	zip
Delete	2	Laura	Juliano	555-1212	Urbandale	IA	50322
Delete	3	Bob	Jones	555-1212	Des Moines	ia	50265

Insert a new Contact

Some items of note from the view:

1. There is a delete link next to each record which triggers a delete servlet and will delete that record in the database before redirecting back to the view
2. There is an 'Insert a new Contact' link at the bottom of the page that directs the user to ContactsInsert.html

The insert page should look similar to:



Insert a contact

First Name: Sally

Last Name: Smith

Phone Number: 123-456

City: Philly

State: PA

Zip: 12345

An item to note about insert: once the user clicks submit, the form directs to a ContactsInsertServlet.java, which insert a record into the database then redirects to the view.

A description of each file needed for this exercise:

1. Write a sql script that creates a Contacts table in derby that holds text values for id, firstName, lastName, phoneNumber, city, state, and zip. Add to the script a couple of INSERT statements so that the table will not be empty. Run your script in Squirrel to confirm it works, and so you have the table on hand for the rest of the exercise. Save the script in the root of the web application as Contacts.sql.
2. ContactsViewServlet.java - displays an html table containing all records in the table. Also displays links to ContactsDeleteServlet.java and ContactsInsert.html.
3. ContactsInsert.html - have text fields for each field in the database. On submit, execution moves to ContactsInsertServlet.java.
4. ContactsInsertServlet.java - uses an INSERT statement to add a record to the database. After the INSERT has been performed, redirect back to ContactsViewServlet.java.
5. ContactsDeleteServlet.java - uses a DELETE statement to records from the database. After the DELETE has been performed, redirect back to ContactsViewServlet.java.
6. web.xml - must have valid references to ContactsViewServlet.java, ContactsInsertServlet.java, and ContactsDeleteServlet.java.

CS - 404 Design and Analysis of Algorithms

Max. Marks : 50 **Min. Marks : 17**

Unit No.	Topics
Unit 1	**Order Analysis:** Objectives of time analysis of algorithms; Big-oh and Theta notations. Master Theorem and its proof, solution of divide and conquer recurrence relations. Searching, Sorting and Divide and Conquer Strategy: Linear Search, Binary Search.
Unit 2	**Searching, Sorting and Divide and Conquer Strategy:** Merge-sort; Quick-sort with average case analysis. Heaps and heap-sort. Lower bound on comparison-based sorting and Counting sort. Dynamic Programming: methodology and examples (Fibonaci numbers, Knapsack problem and some other simple examples) **Dynamic Programming:** Longest integer subsequence, Longest common subsequence ,Weighted interval scheduling.
Unit 3	**Greedy Method:** Methodology, examples (lecture Scheduling, process scheduling) and comparison with DP (more examples to come later in graph algorithms) **Greedy Method:** Knapsack problem and some other simple examples. **Graph Algorithms:** Basics of graphs and their representations. BFS. DFS. Topological sorting.
Unit 4	Minimum spanning trees (Kruskal and Prim'salgorithms and brief discussions of disjoint set and Fibonacci heap data structures). Shortest Paths (Dijkstra, Bellman-Ford, Floyd-Warshall). Hard problems and approximation algorithms. Problem classes P, NP, NP-hard and NP-complete, deterministic and nondeterministic polynomial-time algorithms, Approximation algorithms for some NP-complete problems.
Unit 5	Backtracking, Branch and Bound technique, String Matching, Knave algorithm, KMP algorithm, Parallel Algorithms

1. Text Book :
   Cormen, Leiserson, and Rivest. *Algorithms,* MIT Press 2001
2. Reference books :
   1. Essential References ALGORITHMS IN C++ by Robert Sedgewick (Pearson Education)2008
   2. Fundamentals of Computer Algorithms by Ellis Horowitz and Sartaj Sahni (Galgotia Publication 1998

ASSIGNMENTS- Design and Analysis of Algorithms

Following practical assignments must be done:-

1. **Coverage of Searching** : Sequential search, QuickSequential search, Binary Search, Interpolation Search, Worst case and average case analysis of above mentioned algorithms
2. **Coverage of Sorting**: Divide and conquer strategy, Merge Sort, Quick Sort
Heap sort, Insertion Sort, Selection Sort, Bubble Sort, Shell Sort, Analysis of above mentioned algorithms

3.**Design Strategies:** Dynamic Programming Methodology(Fibonacci numbers, Knapsack problem, Longest integer subsequence, Longest common subsequence ,Weighted interval scheduling

4. **Greedy method:** Greedy method, Knapsack problem, Coin Problem

5. **Graph Algorithms:** Breadth first search, Depth first search, Topological sorting, Minimum weight spanning trees, Prim algorithm, Kruskal algorithm, Dijkstra algorithm, Bellman Ford algorithm

6. **Examples of different problem classes:** Class P, Class NP, NP hard problems, NP complete problems, Deterministic and non deterministic polynomial time algorithms, Approximation algorithms for some NP-complete problems for example approximation algorithms for vertex cover problem and Set cover problem

7. **Programs :** Backtracking, Branch and bound technique, solution of n queen problem and night tour problem using above mentioned concepts, String Matching

8. **Implementation:** Parallel Algorithms

Effective for Students Admitted in July, 2011 onwards