Syllabus for

BACHELOR OF SCIENCE
(COMPUTER SCIENCE Honours)
(Semester System)

Approved by Board of Studies on 5th May 2015

Approved by Academic Council on 12th May 2015

NORTH EASTERN HILL UNIVERSITY
SHILLONG

2015

Preamble

Overview

The Three Year B.Sc. Course in Computer Science, for the semester system builds on the recently revised course of Computer Science for classes VIII to XII. The syllabus tries to give papers that are core to the field of computer science. The choice of the papers is made such way that a student going through the three year degree course can make a clear understanding of the recent areas of computing, while stress is also given to core subjects and fields that are basic to this field.

Eligibility

Students who have satisfied the University norms and if they have taken up Computer Science in Class XII, or had Mathematics in Class XII.

Practical Record Book

In the papers involving practicals, a standard set of problems have been listed. These and/or others similar to them are to be done as practical work and submitted by the student in a laboratory record book. For each problem, the following sections are to be recorded :

1. Definition of the problem
2. Glossary of variables
3. Pseudo code and/or flowchart
4. Sample test data
5. Source code
6. Sample input/output screens.

Internal Assessment

The marks for internal assessment specified for each paper is to be given on the basis of

1. Tests held during the year (for both theory and practical)
2. Assignments submitted and/or seminars given
3. Laboratory record book in the papers applicable.

A Record of this may be maintained in the college.

Practical Examination

For the examination in practicals, the problems need not be restricted to those given in the syllabus. However, they should be of similar standard. For evaluation of practical examination, the following points may be considered:

- Logic (pseudocode and/or flowchart, source code, syntax): Weightage 60%
- Screens: Weightage 10%
- Completion: Weightage 30%

Duration of Examinations:

Theory Papers: Three (3) Hours for papers with 75 marks
Two (2) Hours for papers with 45 marks

Practical Papers: Three (3) Hours

Course Structure

YEAR	SEM	PAPER	CODE	PAPER NAME	Theory		Practical		Total
					External	Internal	External	Internal	
1	1	I	CS101T	Introduction to Data Structure using C	37.5	12.5			100
			CS101P	Introduction to Data Structure using C			37.5	12.5	
	2	II	CS201T	Digital Logic Design and Computer Architecture	75	25			100
2	3	III	CS301T	Database Management System	56	19			100
		III	CS301P	Database Management System			19	06	
	4	IV	CS401T	Data Communication and Computer Networks	75	25			100
3	5	V	CS501T	Operating Systems and Introduction to Linux	56	19			100
			CS501P	Operating Systems and Introduction to Linux			19	06	
		VI	CS502AT	*Option 1:* Computer Graphics	37.5	12.5			100
			CS502AP	*Option 1:* Computer Graphics			37.5	12.5	
			CS502BT	*Option2*: Programming using VB.NET	37.5	12.5			100
			CS502BP	*Option2*: Programming using VB.NET			37.5	12.5	
			CS502CT	*Option 3:* Object Oriented Programming Through Java	37.5	12.5			100
			CS502CP	*Option 3:* Object Oriented Programming Through Java			37.5	12.5	
	6	VII	CS601T	Software Engineering	37.5	12.5			100
			CS601P	Project			37.5	12.5	
		VIII	CS602AT	*Option 1*: Compiler Design	75	25			100
			CS602BT	*Option 2*: Artificial Intelligence	75	25			100
			CS602CT	*Option 3:* Data Mining	75	25			100
				TOTAL	**449.5**	**150.5**	**150.5**	**49.5**	**800**
					(600)		**(200)**		

CS101T and CS101P: Data Structures using C

Objective:

The objective of the course is to introduce the fundamentals of C programming language and develop the skills for solving problems using computers. After completion of this course, a student will be able to

- Understand and use the process of abstraction using a programming language such as 'C'
- Analyse step by step and develop a program to solve real world problems
- Understand File handling in C and the basics of Graphics programming.
- Understanding and implementing Linear and Nonlinear data structures: Arrays, lists, stacks, queues, trees and graphs
- Implementation of various sorting and searching methods and their complexity analysis.

Outline of the Course

Minimum Class Hours			Exam time (Hours)		Marks				
					Theory		Practical		Total
Theory	Practical	Total	Theory	Practical	External	Internal	External	Internal	
60	40	100	2	3	37.5	12.5	37.5	12.5	100

Unit	Topic	Minimum Class Hours			Marks (Theory)
		Theory	Practical	Total	
I	C fundamentals, I/O functions, Control statements, Functions, Arrays and Pointers, Structure and Union	15	10	25	9.5
II	Linear Data Structures: Linked List, Stacks and Queues	10	10	20	6.5
III	Trees	15	7	22	8.5
IV	Graphs	10	8	18	6.5
V	Searching and Sorting, and their complexity analysis	10	5	15	6.5
Total		60	40	100	37.5

Detailed Syllabus

Unit I: C Fundamentals, I/O functions, Control statements, Functions, Arrays and Pointers, Structure and Union **15 Hours**

C Fundamentals: Algorithms, Flow charts, Development of algorithms, The C character set, identifiers and keywords, Data types, constants, variables and arrays, declarations, symbolic constants, Operators (Arithmetic, unary, relational, logical, bitwise, assignment),

I/O functions: Header files (stdio.h, conio.h) getch(), getche(), getchar(), putch(), putchar(), scanf(), printf(), gets(), puts(), clrscr(), window().

Control statements: Decision making and branching (**if..else**, **switch)**, Decision making and looping (**while**, **do .. while**, **for**), Jumping (**break**, **continue**, **goto),** Nested loops.

Functions: Overview (definition, declaration), defining and accessing a function, function prototypes, call by value, call by reference, recursion, Advantages and disadvantages of recursion over iteration, Storage classes (Automatic, Register, External, Static), String functions , Math functions , Memory allocation functions,

Arrays and Pointers: Defining an array, array initialization, processing an array, passing array to a function, multidimensional arrays, arrays and strings, pointer declarations, passing pointer to a function, pointer and one dimensional arrays, Operation on pointers.

Structures and Unions: Defining a structure, processing a structure, user defined data types, structures and arrays, structures and pointers, passing structures to a function, self-referential structures, Union, Union of structures, Enumerated, typedef.

Unit II: Linear Data Structures: Linked List, Stacks and Queues 10 Hours

Data Type, Abstract Data Type, Data Structure, Complexity measures in terms of time and space; Big O notation. Linked List as a data structure (characteristics, advantages, disadvantages); operations on lists (creation, insertion, deletion, traversal, merging, splitting); singly linked list, doubly linked list, circular list; use of linked lists for polynomial representation and manipulation (addition and multiplication).

Stacks and Queues as data structures; implementation of stacks and queues using arrays and linked lists; Definitions of Circular Queue, Priority Queue, D-Queue; Application of stacks : Conversion of infix(containing arithmetic operators including exponential operator, and parenthesis) to postfix, evaluation of postfix expression.

Unit III: Trees 15 Hours

Definition of tree as a data structure (Binary Trees and General Trees), Basic Terms (father, son, descendant, ancestor, height, depth, leaf, node, forest, ordered trees, strictly binary tree, complete binary tree, internal nodes, external nodes); Representation of trees using linked lists, Binary tree traversal methods (pre-order, in-order, post-order), recursive algorithms for traversal methods, Binary search trees (creation, insertion and deletion of a node), Height balanced (AVL) binary trees (construct and traverse an AVL tree), Definitions and characteristics of threaded binary trees, multi-way search trees and B-tree.

Unit IV: Graphs

Definition of a graph, Basic Terms, Graph representation: Adjacency matrix, adjacency lists, incidence matrix, adjacency multi-lists; Traversal schemes: Depth first search, Breadth first search (Recursive and non-recursive algorithms); Shortest Path algorithms (Dijkstra's), Spanning tree, Minimal spanning tree algorithms (Kruskal's algorithm)

Unit V: Searching and Sorting, and their complexity analysis 10 Hours

Linear and binary search and their complexity analysis; Hashing, Hash Functions (division method, mid square method, folding), Analysis of ideal hash function; Conflict resolution (linear and quadratic probe, double hashing, separate chaining, coalesced chaining); Analysis of collision resolution techniques; Sorting algorithms(Insertion, Selection, Bubble, Quick) and comparison of their time complexity.

Instructions For Paper Setter

The question papers will be set according to the following scheme

Cont'd …

Unit	Theory Questions		Practical Questions		
	To be set	To be answered	To be set	To be answered	Marks
I	2	1	2	1	20
II	2	1			
III	2	1			
IV	2	1	2	1	10
V	2	1	2	1	37.5

Distribution of marks for Practical

10% : Syntax and input/output screens
30% : Logic and efficiency (source code, pseudocode, and algorithm)
20% : Error trapping (illegal or invalid input, stack overflow, underflow, insufficient physical memory etc.)
20% : Completion
20% : Result

Recommended Books

***Textbooks*:**

1. **Yashavant Kanetkar**, *Let us C* , BPB Publication
2. **S. Chattopadhyay, D. Ghosh Dastidar, M Chattopdhyay**, *Data Structures Through C Language*, BPB Publications, 2001
3. **William M. Newman, Robert F. Sproull**, *Principles of Interactive Computer Graphics*, Tata McGraw Hill Publishing Co Ltd.,

Reference books:

1. **Byron S. Gottfried**, *Theory and Problems of Programming with C*, Tata McGraw Hill Publication
2. **Hearn & Baker**, *Computer Graphics*, Prentice Hall India, Ltd.
3. **E. Balaguruswamy**, *Programming in ANSI C*, Tata McGraw Hill publication
4. **Y. Langsam, M.J. Augenstein, A.M. Tenenbaum,** *Data Structures Using C and C++*, Second Edition, Prentice Hall of India, 2000

Practical Assignments

(Questions may not be restricted to this list)

1. Write a program to display the message "Welcome to the C programming world" on the screen.
2. Write a program to find out the sum of two integer values and display the result on the screen. Input the two values from the keyboard.
3. Write a program to find out the greatest of three numbers.
4. Write a program for **swap**ping the two numbers with / without using another variable.
5. Write a program to find whether the given year is a leap year or not (use % modulus operator)
6. Write a program to find out the real roots of quadratic equation, $Ax^2+Bx+C=0$.
7. Write a program to convert the given temperature in Fahrenheit to Celsius using the following conversion formula, C=(F-32)/1.8.
8. Write a program to find out the average of any ten numbers. (Use (a) **while** loop, and (b) **for** loop).

9. Write a program to generate Fibonacci sequence. (1,1,2,3,5,8,13, …)
10. An employee is paid 1.5 times the normal rate for every hour beyond 40 hours worked in a week. Write a program to calculate the weekly wage of an employee.
11. Write a program to check whether the given string is palindrome or not.
12. The total distance traveled by a vehicle in t seconds is given by

    Distance = $ut + (at^2)/2$

    Where **u** is the initial velocity (meters per second), **a** is the acceleration (meters per second2). Write a program to evaluate the distance traveled at regular intervals of time, given the values of **u** and **a**. The program should provide the flexibility to the user to select his own time intervals and repeat the calculations for different values of **u** and **a**.
13. For a certain electrical circuit with an inductance L and resistance R, the damped natural frequency is given by

    Frequency $=\sqrt{[(1/LC-R^2/4C^2)]}$

    It is desired to study the variation of this frequency with C (capacitance). Write a program to calculate the frequency for different value of C starting from **0.1** in steps of **0.01**.
14. Write a program to read the following numbers, round them off to the nearest integers and print out the results in integer form:

    35.7 50.21 -23.73 -46.45
15. Given the string "WORDPROCESSING ", write a program to read the string from the terminal and display the same in the following formats:

    (a) WORD PROCESSING (b) WORD (c) W. P.

    PROCESSING
16. Admission to a professional course is subject to the following conditions:
    (a) Marks in mathematics >=60
    (b) Marks in physics >=50
    (c) Marks in chemistry >=40
    (d) Total in all three subjects >=200

    Or

    Total in mathematics and physics >=150

    Write a program to search of admission of students. The user has to enter the marks from the keyboard of the corresponding subjects.
17. Write a program that will read the value of **x** and evaluate the following function

    1 for x>0

    Y= 0 for x=0

    -1 for x<0

    Using

(a) nested **if** statements,
(b) **else if** statements, and
(c) Conditional operator?:

18. Write a program to calculate the monthly telephone bill according to the following rules:
    (a) Rural subscribers:

Upto 250 calls	Free
251 calls to 450 calls	0.60
451 calls to 500 calls	0.80
501 calls to 1000 calls	1.00
above 1000 calls	1.20

(b) Urban subscribers:

Upto 150 calls	Free
151 calls to 400 calls	0.80
401 calls to 1000 calls	1.00
Above 1000 calls	1.20

(c) The rental for urban subscribers depends on the number of calls upto 400 calls the rental will be 200/- and above 400 calls the rental will be 240/-. For rural subscribers the rental is always 200/-.

19. Write a C program to input the Name, City Type (whether Metro or Non-Metro) and Basic Pay of an employee and calculate the salary according to the following rules:
    (a) Dearness allowance (DA)
        (i) Upto Rs. 3500 110% of basic pay
        (ii) Above Rs.3500 90% of the basic pay subject to a maximum of Rs. 3850 (i.e. DA should be at least Rs. 3850.
    (b) House Rent Allowance (HRA) is 15% of the basic pay subject to a maximum of Rs. 800 (i.e. never more than Rs. 800)
    (c) If City is Metro, City Compensatory Allowance (CCA)=800 else if it is Non-Metro, CCA=600.
    (d) Provident Fund (PF) is 12% of the basic pay.
    (**Total Salary**=Basic Pay +DA+HRA+CCA-PF)

    The **output** should be in the following format (Example only)

Example Name	ABCDEF	
Basic Salary		5000
Dearness Allowance		4500
HRA	750	
CCA: Non-Metro	600	
PF	600	
Total Salary		10250

20. Write a program to sum the following series:
    a) The first n natural numbers
    b) The first n odd natural numbers
    c) The first n even natural numbers
21. Write a program to sum the series : 2 * 3 – 3 * 5 + 4 * 7 + to n terms
22. Given a number, write a program using while loop to reverse the digits of the number. For example, the number 12345 should be written as 54321. (**Hint:** Use modulus operator to extract the last digit and the integer division by 10 to get the n-1 digit number from the n digit number.)
23. Write a program for sorting the elements of an array by using Selection sort, Bubble sort, Insertion sort.
24. Write a program to generate positive prime numbers.
25. Write a program to display the multiplication table of a given number from 1 to 20.
26. Write a program to display the multiplication table of a given number for a given range.
27. Write a program to display the multiplication table of a given group of numbers (maximum five numbers) for a given range.
21. Write a program to find the biggest and smallest number and its position in the given array.
22. Write a program to find addition, subtraction and multiplication of matrices using function.
23. The factorial of an integer m is the product of consecutive integers from 1 to m. That is, Factorial m = m! =m*(m-1)*(m-2)*…*1.

24. Write a program to find the sum of row, column, and diagonals of the given matrix.
25. Write a program to find the largest number of the given matrix using function.
26. Write a program to sort all the elements of a matrix using function.
27. Write a program to input a string and perform the following tasks without using library functions: (a) to find its length, (b) to change it to upper case / lower case (c) to extract the left most n characters, (d) to extract the right most n characters (e) to extract n characters from it starting from position p, (f) to insert another string in it at position p (g) to replace n characters in it starting at position p with a given string
28. Write a program to search a pattern in a given text.
29. Write a program to search a pattern in a given text and replace every occurrence of it with another given string.
30. Write a program to write a given number in words using function.
31. Write a program to display the text in a FILE. (TYPE command in DOS).
32. Write a program to copy the contents of one text to another text file using command line arguments.
33. Write a program to merge the two text file to another text file.
34. Write a program to copy the contents of one text file to any number of given files using command line arguments.
35. Write a program to count the number of characters, lines and words in a text file.
36. Write a program to print every line of a text file containing a given pattern.
37. To copy a file by converting lower case text file to upper case text file using command line argument.
38. Write a program to input, sort, and display n names using using array of pointers.
39. Write a program to count the number of vowels, consonants, and other characters and the number of words in a string / file. A space, tab, or a punctuation mark separates a word (, ; . : !).
40. Write a menu driven program to create records of students with marks in various subjects and store them in a file (sequential, random or binary). Make provision for viewing all the records, searching a particular record, editing a particular record, deleting a particular record and listing a particular group of records.
41. Write a menu-driven program to
    a) Construct a singly linked list. Assume the information part of each node consists of only an integer key. Get input for each key from the keyboard. Assume the input is over when the user enters –1
    b) Print the information from each node
    c) Delete all nodes containing a given number
    d) Exit
42. Consider that *L*, a linked list of *n* integers, is given to you. Suppose, the nodes of the list are numbered from *1* to *n*. It is required to split the list *L* into 4 lists so that the first list contains the nodes of *L* numbered 1, 5, 9, 13 … The second list contains the nodes numbered 2, 6, 10, 14 … The third list contains the nodes of *L* numbered 3, 7, 11, 15, …. The fourth list contains the nodes of *L* numbered 4, 8, 12, 16 … Write a program to create the list and perform the splitting.
43. Write a C function to insert a node appropriately to an already sorted list so that after insertion, the new list also becomes sorted. Take care of special cases such as inserting into an empty list. Use this function to write a program which accepts integers at the input and at the end produces a sorted list. Assume that if the integer read at the input is '0' then your program should stop.

44. Write a program to implement polynomial multiplication. Test your program by inputting the following two polynomials given below.
45. 10 P^8 + 14 P^6 - 8 P^5 - 3 P^4 + P^2
46. 3 P^ 4 + 5 P^3 - 2 P + 9
47. (^ is to be read as "raised to")
48. Store each term of the polynomial in a linked list in descending order of the index. Use separate linked lists for each polynomial. Obtain and store the product in a third linked list, and then print out all the three polynomials in a format similar to the one shown above, in descending order of index.
49. A bi-directional list is a list of elements that are linked in both ways. Both links are originating from a header. Construct a module with procedures for searching, inserting and deleting elements.
50. Write a program to represent a sparse matrix using linked list. Add together two such matrices, and display the original and resulting matrices in matrix form.
51. Write a menu-driven program to implement a stack ***using arrays***. The menu should have the following options:
    a. Push on to the stack
    b. Pop from the stack and print the value popped from the stack
    c. Merely print the value on top of the stack
    d. Exit
52. Error trapping should be done for underflow and overflow. Available array space should be efficiently used (i.e. there cannot be overflow if there is more than 1 empty element in the array). Assume that the information part of a stack element is only an integer.
53. Write INSERT and DELETE functions in C language simulating insertion and deletion in circular queue which stores an array of characters.
54. A double-ended queue is a linear list in which additions and deletions may be made at either end of the queue. Write a C function to implement deque with desired functionality. Illustrate use of your function in an example problem, say, a queue of integers.
55. Devise a scheme to traverse a singly linked list in both directions by reversing the links during left to right traversal. Write a C program to implement this traversal scheme.
56. Write a C program to convert an expression from its infix form to its equivalent (a)postfix form, (b)prefix form. Assume the infix expression contains only operators +, -, /, *, ^. The operator '^' stands for exponentiation. The operands are all single digit integers. Display the resulting (a)postfix expression (b)prefix expression.
57. Write a program to input a postfix expression that consists of only single digit positive operands and the binary operators **+, -, *,** and **/.** Using a function, evaluate this postfix expression. The function should report if the postfix expression is invalid, else return its value. [For example**, 242/-46*+7+** is a valid postfix expression (being the equivalent of the infix expression, **2-4/2+4*6+7**) and its value is 31.00.]
58. Write a program to construct a binary search tree of integers using linked list. Assume the information part of each node consists of only an integer key. Get input for each key from the keyboard. Assume the input is over when the user enters –1. Next, print out the keys in ascending order of magnitude, using a non-recursive function.
59. Write a program to create a binary tree and to traverse the tree in
    a. pre-order

b. in-order

c. post-order

60. Implement a procedure for deleting an element *X* from a binary search tree.
61. Write a program to reconstruct a binary search tree given its pre-order and in-order traversal sequence.
62. Write a program to find the biggest and smallest item in a binary search tree.
63. Design and implement an algorithm for insertion of an element in AVL tree taking into account all possible conditions.
64. Represent a graph using adjacency matrix. Write a C procedure to transform an adjacency matrix based representation to a linked-list based representation.
65. Design a suitable representation so that a graph can be stored on a hard disk. Write a procedure adjacency matrix based representation.
66. Write a program to represent a graph and perform a non-recursive depth first search of an item in it.
67. Write a program to represent a graph and compute the shortest distance between two nodes in it.
68. Write a program to input some numbers into an array, and then sort them using various sorting techniques (selection sort, bubble sort, quick sort, radix sort) and compare their time-complexities.
69. Write a program to input some numbers (at least 128 numbers, the more the better) from a file into two arrays A and B. Sort array B. Perform the linear search in array A and binary search in array B for a given number. Repeat these as many times as user decides and compare the time-complexity of the two search methods on the average.
70. Design and implement an algorithm to delete an identifier *X* from a hash table which uses hash function *f* and linear open addressing to resolve collisions. Your deletion scheme must ensure that correct search is possible even after deletion.

CS201T: Digital Logic Design and Computer Architecture

Objective : The objective of this paper is to introduce the student to the concepts of Digital Logic and Boolean algebra and also the concepts of Computer Architecture.

Outline of the Course

Minimum Class Hours	Exam Time	Marks		
		External	**Internal**	**Total**
100	**3 hours**	75	25	100

Unit	Topic	Class Hours	Marks
I	**Digital Logic**	20	15
II	**Boolean Algebra**	20	15
III	**Combinational Logic, Arithmetic**	20	15
IV	**Sequential Logic, Control Unit**	20	15
V	**Input-Output and Memory Organization**	20	15
Total		**100**	**75**

Detailed Syllabus

Unit I: Digital Logic **20 Hours**

Block Diagram of a digital computer. Bit, Byte, Word. Binary, Decimal, Octal and Hexadecimal Number Systems. AND, OR, NOT, XOR, X-NOR gates, and Buffer (gate) - with Graphic Symbols, Truth Tables, and Algebraic functions. Listing the Truth Table for a given Boolean Expression. Deriving the algebraic expression for the complement of a given Boolean function. Drawing the logic diagram by using gates to represent a given Boolean expression. Two Universal gates-NAND and NOR gates.

Conversion of integers and fractions from one number system to another (viz. Binary, Decimal, Octal, Hexadecimal number systems). 9's, 10's, 1's and 2's Complement. Binary Addition. Subtraction of decimal numbers using 9's and 10's complement. Binary Subtraction of unsigned numbers using 1's and 2's complement.

Unit II: Boolean Algebra **20 Hours**

Canonical Forms - *viz*. Minterms, Maxterms. Standard Forms- viz. Sum of Products, Product of Sums. Simplification of Boolean expressions by using Basic Identities of Boolean Algebra, De Morgan's Theorem.

Drawing the logic diagram by using gates to represent the simplified Boolean expression. Drawing the logic diagram by using only one type of universal gates to represent a given Boolean expression- either only NAND gates or only NOR gates.

Karnaugh-Map (2 variable maps, 3 variable maps, 4 variable maps) method for simplifying Boolean Expressions, Don't Care Conditions.

Unit III: Combinational Logic, Arithmetic **20 Hours**

Half-Adder and Full Adder- with Truth Tables, Logic Diagrams and Using Karnaugh maps to derive the Boolean expressions for the Sum and Carry. Logical Left Shift, Logical Right Shift and Arithmetic Left Shift, Arithmetic Right Shift. Flowchart of Booth's Multiplication Algorithm for multiplying both

positive and negative binary integers in Two's Complement form, Hardware for Booth's Algorithm, The Registers used to store the Multiplicand, Multiplier, Product.

Unit IV: Sequential Logic, Control Unit 20 Hours

Sequential circuits. Block Diagram of a Sequential circuit. Flip flop. Present State and Next State of a flip flop. SR, JK, D and T flip flops using graphic symbols (Characteristic Tables and Excitation Tables). Indeterminate State of an RS flip flop.

Design of counters using SR, JK, D and T flip flops- Present State and Next State in the design of the counter, State Diagram, State Table, Excitation Table of the counter, Use of Karnaugh maps to derive the corresponding flip flop input- equations for the combinational circuit of the counter, and Logic Diagram of the counter.

Brief definition of registers in a Basic computer. Instruction code, op code (operation code), mode field, effective address. Phases of an instruction cycle (Fetch, decode, execute). Block diagram of a CPU, Addressing modes (Immediate, Implied, register, register indirect, direct, indirect, relative, indexed)

Unit V: Input-Output and Memory Organization 20 Hours

Input-Output Organization:

I/O Bus versus Memory Bus, Isolated I/O method versus Memory-Mapped I/O; Strobe Control, Handshaking. Three Modes of Transfer (Programmed I/O, Interrupt-Initiated I/O, DMA).Programmed I/O (Block Diagram of Data Transfer from I/O Device to CPU via an Interface, Flowchartfor CPU Program to input data).

Memory Organization:

Primary memory and Secondary memory. Block Diagrams of RAM and ROM. Magnetic Disks- Tracks, Sectors. Comparison of the Speed, Size and Price of the Primary Memory and Auxiliary Memory.

Cache Memory, Locality of Reference, Hit, Miss, Hit Ratio, Associative memory- match logic, list of Boolean functions related to match logic. Three Mappings in cache viz. Associative Mapping, Direct Mapping (Tag Field, Index Field), Set Associative Mapping (two-way set).

Instruction for Paper Setter

(The question will be set according to the following scheme)

Unit	Topic	To be set	To be answered
I	Digital Logic	2	1
II	Boolean Algebra	2	1
III	Combinational Logic, Arithmetic	2	1
IV	Sequential Logic, Control Unit	2	1
V	Memory Organization	2	1
Total		**10**	**5**

Recommended Books

***Textbooks* :**

1. **M. Morris Mano**, *Computer System Architecture*, Pearson Education, New Delhi, Third Edition, 2007.
2. **M. Morris Mano**, *Digital Logic and Computer Desig*n, Pearson Education, New Delhi

Reference books:

1. M Morris Mano, Digital Design, Pearson Education, New Delhi, Third Edition, 2002
2. Floyed and Jain, Digital Fundamentals, Pearson Education, New Delhi.

CS301T: Database Management System

Objective

The objective of this paper is to introduce to the students the fundamental concepts necessary for designing, using and implementing database systems and applications. The paper stresses on database modeling and design, physical file storage techniques and language facilities provided by database management systems.

Minimum Class Hours	Exam Time (Hours)	Marks		
		External	Internal	Total
60	2	56	19	75

Unit	Topic	Minimum Class hours	Marks
I	Introduction and Conceptual Data Modeling	13	15
II	File Organization	13	08
III	Relational Data Model	13	15
IV	Functional Dependencies and Normalization	8	06
V	Transaction Processing, Concurrency Control, Security	13	12
	Total	**60**	**56**

Detailed Syllabus

Unit I: Introduction and Conceptual Data modeling 13 Hours

Introduction: Introduction to databases, characteristics of the database approach, role of a DBA, advantages of using a DBMS, data models, schemas, instances, DBMS architecture (Three-Schema Architecture), Data Independence

Conceptual Data Modeling: entity type, entity set, attributes, keys, value sets, relationships, relationship types, relationship sets, relationship instances, relationship degree, role names, recursive relationships, constraints on relationship types, attributes of relationship types, weak entity types, ER Diagram, naming conventions and design issues, EER (Concepts only) – Subclass, Superclass and Inheritance, specialisation and generalisation, constraints on specialisation and generalisation

Unit II: File Organization 13 Hours

Introduction to storage hierarchies, hardware descriptions of disk devices, Magnetic Tape Storage Devices, RAID technology, Organization of file records on disk (record and record types, Fixed- length records, variable-length records, record blocking, spanned and unspanned records, allocating file blocks on disk, file headers), primary methods of file organization -Heap Files, Sorted Files, Hashed Files – internal and external hashing, dynamic file expansion. Types of Single-level Ordered Indexes (Primary Indexes, Clustering Indexes, and Secondary Indexes), Multilevel Indexes: Basic technique

Unit-III: Relational Data Model 13 Hours

Relational model concepts: Domain, attribute, tuple, relation, characteristics of relations, relational databases, relational database schemas, relational constraints (Domain constraint, constraints on null), entity integrity, referential integrity, foreign keys. ER to Relational mapping algorithm.

Relational Algebra: basic relational algebra operations – SELECT, PROJECT, UNION, INTERSECTION, SET DIFFERENCE, Cartesian PRODUCT, JOIN, Division, Aggregate functions (SUM, COUNT, MAXIMUM, MINIMUM, AVERAGE)

SQL: Characteristics of SQL, Data types in SQL, Types of SQL commands
Data Definition Commands: CREATE SCHEMA, CREATE TABLE, DROP TABLE, ALTER TABLE
Data changing commands: INSERT, DELETE, UPDATE
Single table query commands: SELECT, SELECT with WHERE, SELECT with ORDER BY, SELECT with GROUP BY, SELECT with GROUP BY and HAVING, SQL built-in functions - SUM, MIN, MAX, COUNT, AVG
Multi-table query commands: Retrieval using sub-query, JOIN, EXIST and NOT EXIST
Special operators: IS NULL, IS NOT NULL, BETWEEN..AND, IN, LIKE, ANY, ALL
Views – CREATE VIEW, DROP VIEW

Unit IV Functional Dependencies and Normalization 8 Hours

Functional Dependencies, First Normal Form, Second Normal Form, Third Normal Form, Boyce-Codd Normal Form, Multivalued Dependencies, Fourth Normal Form

Unit V: **Transaction Processing, Concurrency Control and Security 13 Hours**

Transaction Processing: Introduction, ACID properties of transaction, transaction states, schedules, serializability, tests for serializability, recoverability
Concurrency Control: Concurrent execution of transaction, Lock-based techniques for concurrency control – Two-Phase locking protocol and its variations, Timestamp based protocol, Deadlock, Deadlock prevention protocols, Deadlock detection, Starvation
Security: Database security issues, threats to databases, role of the DBA in maintaining database security, access protection, encryption, database audits

Instructions to Paper Setter

Questions should be set according to the following scheme.

Unit	Questions	
	To be set	To be answered
I	2	1
II	2	1
III	2	1
IV	2	1
V	2	1

Recommended Books

***Textbooks*:**

1. **R. Elmasri, S.B Navathe**, *Fundamentals of Database Systems*, Addision, Wesley (Fifth Edition)

Reference books:

1. **A. Silberschatz, H.F Korth, S Sudarshan**, *Database System Concepts*, ***Tata***- McGraw Hill, 1997.
2. **Bipin Desai**, *An Introduction to Database Systems*, Galgotia Publications (West Publishing), 1991.
3. **D.M Kroenke**, *Database Processing: Fundamentals, Design and Implementation*, Prentice-Hall of India, (Eighth Edition) 2002.
4. **G.W Hansen, J.V Hansen**, *Database Management and Design*, Prentice-Hall of India, (2nd Edition) 2001.
5. **Thomas M Connolly, Carolyn E Begg,** *Database Systems, A Practical Approach to Design, Implementation and Management*, Addison Wesley Longman Ltd. 1999.
6. **R. Panneerselval,** *Database Management Systems*, PHI, second edition, 2011.

CS301P: Database Management System

Objective

The objective of this paper is to make the students implement the concepts of a database system practically. Any RDBMS software may be used for the purpose.

Minimum Class Hours	Exam Time (Hours)	Marks		
		External	Internal	Total
40	2	19	06	25

Lab Exercises (SQL Exercises)

1. Exercises on the different data types in SQL and Data definition language (Creating Databases, creating tables with provision to specify various constraints – Primary keys, foreign keys, uniqueness constraint, constraints on nulls)
2. Exercises on Data manipulation language (Queries that involve various joins, special operators, aggregate functions, additional clauses : GROUP BY, ORDER BY and HAVING, Corelated and uncorelated nested queries, insert, delete, update)
3. Exercises on views.

Instructions to Paper Setter

Questions should be set according to the following scheme.

Questions		Marks
To be set	To be answered	**19**
2	1	

Distribution of marks for practical

20% : Database structure
20% : Constraints Specification
40% : Queries
10% : Completion
10% : Result

Recommended Books

Textbooks:

1. **R. Elmasri, S.B Navathe**, *Fundamentals of Database Systems*, Addision, Wesley (Fifth Edition)

Reference books:

1. **Pranab Ghosh,** *SQL Popcorn*, PHI Learning, 2010
2. **A. Silberschatz, H.F Korth, S Sudarshan,** *Database System Concepts***,** Tata- McGraw Hill, 1997**.**
3. **Bipin Desai,** *An Introduction to Database Systems*, Galgotia Publications (West Publishing), 1991.
4. **D.M Kroenke,** *Database Processing: Fundamentals, Design and Implementation*, Prentice-Hall of India, (Eighth Edition) 2002.
5. **Thomas M. Connolly, Carolyn E Begg,** *Database Systems, A Practical Approach to Design, Implementation and Management*, Addison Wesley Longman Ltd. 1999.

CS401T: Data Communications and Computer Networks

Objective:

Computer networks and Web Technology have become topics of utmost significance to the present day information based society. This paper intends to provide students an understanding to the basic concepts and mechanisms underlying computer networks and web technology.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Marks		
		External	Internal	Total
60	2	75	25	100

Unit	Topic	Minimum Class Hours	Marks
I	Introduction to Computer Networks, Physical and Data Link Layer	17	18
II	Network Layer	12	15
III	Transport Layer	11	14
IV	Application Layer and Introduction to Internet	10	14
V	The World Wide Web	10	14
Total		**60**	**75**

Detailed Syllabus

Unit I: Introduction To Computer Networks, Physical and Data Link Layer **17 hours**

Introduction to computer networks

Uses of computer Networks – Business Applications, Home Applications, Mobile users; Network topology – Linear, Bus, Ring, Star, and Hybrid Topology; Net Work Hardware – LAN, MAN, WAN; Network Software- protocol Hierarchies, Connection Oriented and connectionless services; Reference models-OSI Reference model, TCP/IP Reference model, Wireless LANSs: 802.11

Physical Layer

Guided Transmission Media- Twisted pair, Fibre Optics; Wireless transmission- Radio Transmission, Microwave Transmission, Frequency Division and Time division Multiplexing; Switching-circuit switching, message switching, packet switching;

Data Link Layer

Services provided to the Network layer, Framing, Error control, Flow control, Error-detecting codes; Simplex stop-and-wait Protocol; A simplex protocol for a noisy channel; Sliding window protocols- A go Back N and selective repeat protocols (outline)

Unit II: Network Layer **12 hours**

Design Issues – Store-and-forward Packet Switching, Implementation of Connectionless Service, Implementation of Connection-Oriented Service; Routing Algorithms- Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Routing for Mobile Hosts, Routing in Ad Hoc Networks; Congestion Control Algorithms – General Principles, Congestion Control in Virtual-Circuit Subnets, Congestion Control in Datagram Subnets, Load Shedding, Jitter Control; Quality of

Service – Buffering, Leaky Bucket Algorithm, Token Bucket Algorithm; Internet Protocol – IP addresses, Subnets.

Unit III: Transport Layer **11 hours**

Transport Service Primitives; Elements of Transport Protocols – Addressing, Connection Establishment, Connection Release, Flow Control and Buffering, Multiplexing, Crash Recovery.
Internet Transport Protocols – UDP – Remote Procedure Call; TCP – TCP Service Model, Protocol, Header, Connection establishment, Connection Release, Transmission Policy, Congestion Control

Unit IV: Application Layer **10 hours**

Origins, What is the Internet? , Internet Protocol Addresses, Domain Names; Electronic Mail – Architecture and Services, User Agent, Message Transfer – SMTP, Message Delivery – POP3 and IMAP, Telnet, FTP, HTTP

Unit V: Introduction to Internet and the Web **10 hours**

Origins, Web or Internet, Web Browsers, Web Servers: Web Server Operation, General Server Characteristics, Apache, IIS, Uniform Resource Locators: URL Formats, URL Paths, World Wide Consortium (W3C): What is W3C, W3C Recommendations

Instruction to Paper Setter

(The question papers will be set according to the following scheme)

Unit	Questions		Marks
	To Be Set	To Be Answered	
I	2	1	18
II	2	1	15
III	2	1	14
IV	2	1	14
V	2	1	14
Total	10	5	**75**

Recommended Books

***Textbooks*:**

1. **Andrew S. Tanenbaum**, *Computer Networks (Fifth Ed.)*, Prentice Hall of India
2. **Uttam K. Roy**, *Web Technologies (1st edition)* , Oxford University Press
3. **Robert W. Sebesta**, *Programming with World Wide Web (4th edition)*, Pearson Education

Reference books:

1. **Sudakshina Kundu**, *Fundamentals of Computer Networks*, Prentice Hall of India , New Delhi, 2005
2. **Behrouz A. Forouzan**, *Data Communications and Networking(Eight Ed.)*, Tata McGraw Hill Edition
3. **Deitel, H. M.; P. J. Deitel,** *Internet and World Wide Web: How to Program (Second Edition)*, New Delhi: Prentice-Hall India

CS501T : Operating Systems and Introduction to Linux (Theory)

Objective:
The main objective of this paper is to introduce the students to a layer of software called the Operating System, whose job is to manage all the devices of a computer system and provide user programs with a simple interface to the hardware. This paper will familiarize the students with the concepts of processes, memory management, file management, Input/Output management and the potential problem of deadlocks.

Minimum Class Hours	Exam Time	Marks		
		External	Internal	Total
70	2	56	19	75

Unit	Topic	Number of Class Hours	Marks
I	**Concepts & Processes**	14	12
II	**Deadlocks**	14	11
III	**Memory Management**	14	11
IV	**File Systems and I/O**	16	11
V	**Linux & Shell Programming**	12	11
Total		**70**	**56**

Detailed Syllabus

Unit I: Concepts & Processes **14 hours**

Operating System objectives and functions, Booting the computer, Operating System concepts (Processes, Address Spaces, Files, I/O, The Shell), System Calls for Process Management, Types of Operating Systems (Mainframe Operating Systems, Multiprocessor Operating systems, Personal Computer Operating systems, Embedded Operating systems, Real Time Operating systems), Introduction to Processes (The Process Model, Process Creation, Process Termination, Process Hierarchies, Process States, Implementation of Processes), Interprocess Communication (Race conditions, Critical Regions, Mutual Exclusion with Busy Waiting (Lock variables, Peterson Solution, TSL instruction), Sleep and Wakeup, Semaphores, Message Passing), Process Scheduling (First come First serve, Shortest Job First, Priority Scheduling, Priority Scheduling, Multiple Queues, Round Robin Scheduling), Classical IPC problems (The Dining Philosophers Problem, The Sleeping Barber Problem).

Unit II: Deadlocks **14 hours**

Resources, Introduction to Deadlocks (Conditions for Deadlock, Deadlock modeling), Deadlock detection and recovery (Deadlock detection with one resource of one type, Deadlock detection with multiple resources of each type, Recovery from Deadlock), Deadlock avoidance (Resource trajectories, Safe and unsafe states, the Banker's algorithm for a single resource), Deadlock prevention (attacking the mutual exclusion condition, attacking the hold and wait condition, attacking the no-preemption condition, attacking the circular-wait condition).

Unit III: Memory Management **14 hours**

Memory Management without Abstraction, A Memory Abstraction(address spaces, Swapping, Managing Free Memory), Virtual Memory (Paging, Page Tables), Page Replacement Algorithms (Not-recently-used, First in first out, Second Chance page replacement algorithm, Least Recently used page replacement algorithm), Design issues for Paging Systems (Page size, shared pages, Mapped Files), Implementation issues(Operating System involvement with Paging, Page Fault Handling, Locking pages in Memory), Segmentation (Implementation of pure segmentation, Segmentation with Paging: MULTICS).

Unit IV: File Systems and I/O Management **16 hours**

Files (File Naming, File structure, File types, File access, File attributes, File operations), Directories (Hierarchical directory systems, Path names, Directory operations), File System Implementation (File system Layout, Implementing Files, Implementing directories, Shared files, Disk space management, File system Backups, File system Consistency, File system Performance)

Principles of I/O hardware (I/O devices, Device Controllers, Direct memory access), Goals of the I/O software, I/O software Layers (Interrupt Handlers, Device Drivers, Device independent I/O software, User space I/O software), Disks (Disk hardware (Magnetic disks, RAID), disk arm scheduling algorithms, Error handling), Clocks (Clock hardware, Clock software).

Unit V: Introduction to Linux and Shell Programming **12 hours**

File Systems and Architecture; The Shell (sh command, Pattern Matching, escaping, quoting, Redirection, /dev/null, /dev/tty, Pipes, command substitution, shell variables)
Simple Filters (pr, head, tail, cut, paste, sort, uniq, tr)\

Instruction for Paper Setter

Unit	Questions	
	To Set	**To Answer**
I	2	1
II	2	1
III	2	1
IV	2	1
V	2	1
Total	**10**	**5**

Recommended Books

Text books:

1. **Andrew S Tanenbaum**, *Modern Operating Systems*, (Third Edition), Pearson Education.

Reference books

1. **William Stallings**, *Operating Systems: Internals and Design Principles*, (Sixth Ed.), Pearson Education, New Delhi.

2. **D.M Dhamdhere,** *Operating systems – A concept based approach*, Second Edition, Tata McGraw Hill 2002

3. **Abraham Silberschatz and Peter B. Galvin**, *Operating System Concepts*, Fourth edition, Addison-Wesley.

CS501P : Operating Systes and Introduction to Linux (Practical)

Objective:
The paper aim at helping the students to learn the various Unix/Linux commands while working in the Unix/Linux environment. The paper also aims in teaching the students to write shell script programs to perform some computations and handling the various unix commands through scripts. This is a practical-oriented paper.

Minimum Class Hours	Exam Time	Marks		
30	3	External	Internal	Total
		30	10	40

Unit	Topic	Number of Class Hours	Marks
I	**Understanding unix commands, general purpose utilities, navigating the file system**	06	**19**
II	**Handling Ordinary Files, Basic File Attributes, More File Attributes, The Vi editor**	06	
III	**The Shell, Simple Filters,**	06	
IV	**Regular Expressions ,The Process**	06	
V	**Shell Programming**	06	
Total		**30**	**19**

Detailed Syllabus

Unit I **06 hours**

Understanding Unix Commands (Internal and External Commands, Arguments, Options and Filenames, Flexibility of Command Usage (Combining Commands & Splitting the Command Line), man pages(Navigating & Searching, brief description of Sections)
General-Purpose Utilities: cal, date, calendar, who, tty, uname, passwd, lock, echo, tput, bc, script, spell and ispell
Navigating the File System (The Three types of Files, Parent-Child relationship, pwd, absolute pathnames, Using absolute pathname for a command, cd, mkdir, rmdir, ls, relative pathnames)

Unit II **06 hours**

Handling Ordinary Files (cat, cp, rm, mv, more, lp, file, wc, od, split, cmp, comm, diff)
The ls Command (ls –l, -d, -x, -a, -R option)
Basic File Attributes (File permissions, chmod)
More File Attributes (File systems, inodes, chown, listing by modification and access times, touch, ln)
The Vi Editor (Three modes, Input Mode, Saving text and quitting, Command mode, the repeat factor, deletion, navigation, pattern search, joining lines, repeating the last command, undoing last editing instructions, search and replace)

Unit III **06 hours**

The Shell (sh command, Pattern Matching, escaping, quoting, Redirection, /dev/null, /dev/tty, Pipes, command substitution, shell variables)
Simple Filters (pr, head, tail, cut, paste, sort, uniq, tr)

Unit IV **06 hours**

Regular expressions (grep, Regular expression, egrep)

The Process (The Shell process, parents and children, ps, system processes, mechanism of process creation, running jobs in background, kill, nice)

Unit V **06 hours**

Shell Programming (Shell scripts, read, command line arguments, exit, logical operators, if, case, expr, sleep and wait, while, until, for, $@)

Recommended Book:

1. Yashvant P. Kanetkar, *Unix Shell Programming*, BPB Publications

Reference Books:

1. **Sumitabha Das**, *UNIX Concepts & Applications*, (Second Ed.), Tata McGraw Hill, 2001
2. **Newham Cameron,** *Learning the Bash Shell*, (Third Ed), O'Rielly

Instructions for Paper Setter: This paper is purely practical-oriented

Unit	To be set	To be Answered	Marks
I, II, III	3	2	12
IV, V	**2**	1	07

Sample Shell Programs

1. Write a script that will accept two file names from the command line, copy the first file to the second file and then display the contents of the combined file. Proper error message should be displayed in case the copy is not successful.
2. Write a script that will read a filename from the command line and will change the name of the file to filename.aa1 where aa1 is the login_name of the user. (E.g. if the filename is Lucky and the user's login_name is harry then, the filename will be changed to Lucky.harry).
3. Peter's basic salary is input through the keyboard. His dearness allowance is 40% of his basic salary, and house rent allowance is 18% of the basic salary. Write a script to calculate his gross salary.
4. The distance between two cities (in km.) is input through the keyboard. Write a script to convert and print this distance in meters, feet, inches and centimeters.
5. The length and breadth of a rectangle and radius of a circle are input through the keyboard. Write a script to calculate the area and perimeter of the rectangle as well as the area and circumference of the circle.
6. If a five digit number is input through the keyboard, write a script to calculate the sum of its digits.
7. Write a script that accepts a string inputted though the keyboard and echoes a suitable message if it does not have at least 10 characters.
8. In a company, an employee is paid as follows: If his basic salary is less than Rs. 5000, then HRA = 10% of basic salary and DA = 90% of basic. If his salary is either equal or above Rs. 5000, then HRA = Rs. 900 and DA = 98% of basic salary. If the employee's salary is input through the keyboard, write a script to find his gross salary.
9. Write a script that will accept a filename from the keyboard and determine whether the file exists. If the file exists then is contents will be displayed else an error message will be displayed.
10. Write a script that accepts two directory names as arguments and deletes those files in the second directory that are identical to the files in the first.
11. The marks obtained by a student in five different subjects are input through the keyboard. The student gets a division as per the following rules:

i. Percentage above or equal to 60 – First division
ii. Percentage between 50 and 59 – Second division
iii. Percentage between 40 and 49 – Third division
iv. Percentage less than 40 – Fail

Write a script to find the division obtained by the student.

12. Write a shell script that will prompt the user to enter a character. The script will then determine whether the user entered a lowercase letter, an uppercase letter, a digit or a special symbol.
13. If the cost price and selling price of an item is input through the keyboard, write a script to determine whether the seller has made profit or incurred loss. Also determine how much profit was made or loss incurred.
14. An integer is input through the keyboard. Write a script to find out whether it is an odd or even number.
15. Write a shell script, which receives any year from the keyboard and determines whether the year is leap or not. If no argument is supplied the current year should be assumed.
16. Write a shell script that will display all the multiples of 5 between 5 and 100.
17. Write a shell script that will display the multiplication table of any given number.
18. Write a shell script to find the factorial of any number entered through the keyboard.
19. Write a script to calculate overtime pay of 10 employees. Overtime is paid at the rate of Rs. 12 per hour for every hour worked above 40 hours. Assume that employees do not work for fractional part of an hour.
20. Write a script that converts a decimal number to a hexadecimal number. [Hint: use bc]
21. Two numbers are entered through the keyboard. Write a script to find the value of one number raised to the power of another.
22. Write a script to print all prime numbers between 1 and 150.
23. Write a shell script that displays a list of all files in the current directory to which you have read, write and execute permissions.
24. Write a shell script that will receive any number of filenames as arguments. The shell script should check whether every argument supplied is a file or a directory. If it is a directory it should be appropriately reported. If it is a filename then name of the file as well as the number of lines present in it should be reported.
25. Write a script that will receive any number of filenames as arguments. The script should check whether such files already exist. If they do, then it should be reported. If these files do not exist then check if a sub-directory called *mydir*exists in the current directory. If it doesn't exist then it should be created and in it the files supplied as arguments should get created. If *mydir*already exists then it should be reported along with the number of files that are currently present in *mydir*.
26. Write a script that accepts a filename as argument and displays the last modification time if the file exists, and a suitable message if it doesn't.
27. Write shell script to convert file names from UPPERCASE to lowercase file names or vice versa.
28. Write a shell script that backs up all files in a directory into a Backup directory for every day of the week. In other words, on Monday all files go in a "Monday" or "1" backup directory, on Tuesday they all go into a "Tuesday" directory, and so forth. If a directory for today already exists, overwrite the files in it, otherwise create the directory.
29. Write a program that checks if any of a list of users given on the command line is logged in. For each user it should say whether he/she is logged in or not.

****

CS502AT and CS502AP: Computer Graphics

Objective: Computer graphics is one of the most exciting and rapidly growing computer field. It has got numerous areas of applications such as user interface, data visualization, television commercials, motion pictures etc. This paper is meant to give the students knowledge of hardware, graphics concept and algorithms to implement the concepts

Outline of the Course

Minimum Class Hours			Exam time (Hours)		Marks				
					Theory		Practical		Total
Theory	Practical	Total	Theory	Practical	External	Internal	External	Internal	
60	40	100	2	3	37.5	12.5	37.5	12.5	100

Unit	Topic	Minimum class hours		Marks	
		Theory	Practical	Theory	Practical
I	Overview of Graphics Systems	06		08	
II	Output Primitives	16	12	08	37.5
III	Two- Dimensional Geometric Transformations	16	12	08	
IV	Two- Dimensional Viewing	07	6	05	
V	Three Dimensional Concept and Some Object Repre-sentation	15	10	8.5	
Total		60	40	37.5	37.5

Detailed Syllabus

Unit I **06 Hours**

***Overview of Graphics Systems* :** Video Display Devices, Refresh cathode-ray Tubes, Raster Scan Display, Random Scan Display Color CRT Monitor, Direct View Storage Tubes, Flat panel Display, Three Dimensional Viewing Devices, Stereoscopic and Virtual-Reality Systems, Raster Scan Systems Video Controller, Raster Scan Display Processor, Random-Scan Systems, Graphics Monitors and Workstations, Input Devices, Hard Copy Devices, Graphics Software, Coordinate Representations,Graphics Functions, Software Standards, PHIGS Workstations.

Unit II **16 Hours**

***Output Primitives*:** Points and Lines, Line Drawing Algorithms, Loading the Frame Buffer, Line Functions, Circle –generating Algorithms, Ellipse-generating Algorithms, Other Curves, Conic Sections, Polynomial and Spline Curves, Parallel Curve Algorithms, Curve Functions, Pixel Address and Object Geometry Screen Grid Coordinate, Maintaining Geometric Properties of Displayed Objects, Filled-Area Primitives, Scan-line polygon Fill Algorithm, Inside Outside Test, Scan –Line Fill of Curved Boundary Areas, Boundary Fill Algorithm, Flood Fill Algorithm, Fill-Area Functions, Cell Array, Character Generations.

Unit III **16 Hours**

***Two-Dimensional Geometric Transformations* :** Basic Transformations: Translations, Rotations, Scaling; Matrix Representations and Homogeneous Coordinates, Composite Transformations:

Translations, Rotations, Scaling, General Pivot Point Rotations, General Fixed Point Scaling, General Scaling Directions, Concatenation Properties, General Composite
Transformations and Computational Efficiency, Other Transformations: Reflections, Shear; Transformations Between Coordinate Systems, Affine Transformations, Transformation Functions, Raster Method for Transformations.

Unit IV **07 Hours**

***Two-Dimensional Viewing* :** The Viewing Pipeline, Viewing Coordinate Reference Frame, Window-to-Viewport Coordinate Transformations. Two -Dimensional Viewing Functions, Clipping Operations, Point Clipping, Line Clipping: Cohen-Sutherland Line Clipping, Liang-Barsky Line Clipping, Nicholl-Lee-Nicholl Line Clipping; Polygon Clipping: Sutherland-Hodgeman Polygon Clipping, Weiler-Atherton Polygon Clipping; Curve Clipping, Text Clipping, Exterior Clipping.

Unit V **15 Hours.**

***Three Dimensional Concept and Some Object Representation*:** Three-Dimensional Display Methods, Parallel Projections, Perspective Projections, Depth Cueing, Visible Line and Surface, Identification, Surface Rendering, Exploded and Cutway Views, Three-dimensional and Stereoscopic Views, Three Dimensional Graphic Packages, Polygon Surfaces, Polygon Tables, Place Equations, Polygon Meshes, Curved Line and Surfaces, Quadric Surfaces: Sphere, Ellipsoid, Torus, Superquadrics, Superellipse, Superellipsoid, Blobby Objects, Spline Representations, Interpolations and Approximations Splines, Parametric Continuity Conditions, Geometric Continuty Conditions, Spline Specifications, Cubic Spline Intepolation Methods, Natural Cubic Splines, Hermite Interpolations, Cardinal Splines, Kochanek-Bartels Splines, Bezier Curves, Properties of Bezier Curves, Design Technique Using Bezier Curves, CubicBezier Curves, Bezier Surfaces.

Instruction to the paper setter

Units	**Theory**		**Practical**	
	To be set	**To be answered**	**To be set**	**To be answered**
I	**2**	**1**		
II	**2**	**1**	**2**	**1**
III	**2**	**1**		
IV	**2**	**1**		
V	**2**	**1**		

Recommended Books

Text Book:

Hearn, D.; M. P. Baker, *Computer Graphics (*Second Edition**),** New Delhi: Prentice-Hall India

Reference Books:

1. Plastock, R.; G. Kalley, *Theory and Problems of Computer Graphics* (Second Edition), Schaum's Series, New Delhi: Tata McGraw-Hill
2. **Foley, l. J.; A. V. Dam; S. Feiner; J. Huges,** *Computer Graphics : Principles and Practice,* New Delhi: Addison Wesley, 1996
3. **Rogers, D.; J. Adams,** *Mathematical Elements for Computer Graphics* (Second Edition), New Delhi: Tata McGraw-Hill

CS502BT and CS502BP: Programming using VB.NET

Objective: Visual Basic.NET is the latest version of Visual Basic, the most significant evolutionary change yet in the language. At its heart is the totally new .NET framework, a rich and powerful set of classes that provides support for just about any imaginable area of programming – desktop, Internet, database, and so on. The intent of this course is to teach:

- The language Visual Basic,
- The .NET framework,
- Programming logic,
- Database programming, and
- Web application

Outline of the Course

Minimum Class Hours			Exam time (Hours)		Marks				
					Theory		Practical		
Theory	Practical	Total	Theory	Practical	External	Internal	External	Internal	Total
60	40	100	2	3	37.5	12.5	37.5	12.5	100

Unit	Topic	Minimum Class Hours			Marks (Theory)
		Theory	Practical	Total	
I	Language Fundamental, Forms and Controls	10	10	20	08
II	User-Defined Data Types, Procedures & Exception Handling	10	10	20	08
III	Object Oriented Programming	16	08	24	08
IV	MDI Applications & Library Functions	08	04	12	05
V	Database and Web Programming	16	08	24	8.5
Total		**60**	**40**	**100**	**37.5**

Detailed Syllabus

Unit I: Language Fundamentals, Forms & Controls **10+10 Hours**

Getting Started : Types of VB.NET projects ; IDE Components- IDE Menu, Toolbox window, Solution Explorer, Properties Window, Output window, Command window, Tasklist window ; Common Properties of Controls – Name, Font, Enabled, Size, Tag, Text, TabStop, TabIndex ; Common Events – Click, DoubleClick, Enter, Leave, MouseEnter, MouseLeave ; Common Methods- Focus, Clear, Hide, Show, Scale; Building a Console application

Language Fundamentals: Comments, Variables – Declaring variables, Types of variables, Data type identifiers, Strict and Explicit options, Object variables, Converting variable types, widening and narrowing, User-defined data types, Examining variable types, Scope and lifetime of a variable;

Constants, Flow control statements – Test Structures – If…Then, If…The…Else, Select Case, Loop structures – For…Next, Do…Loop, While…End While; Nested Control structures; Exit statement

Working with Forms: The Appearance - Properties of form Controls, Placing Control on the Forms, Setting the Tab Order, Anchoring and Docking, The Form's Events; Loading and Showing Forms - The Startup Form, Controlling one form within another, Forms versus Dialog Boxes; Building Dynamic Forms at Runtime.

Basic Windows Controls - Label, TextBox, CheckBox, RadioButton, ListBox, CheckedListBox, ComboBox, GroupBox, TabContol, Timer, StatusBar, ImageList, PictureBox, HScrollBar and VScrollBar;

Unit II: User-Defined Data Types, Procedures & Exception Handling **10+10 Hours**

Arrays, Structures and Enumerations: Arrays-Declaring and Initializing arrays, array limits, multi-dimensional arrays, dynamic arrays, arrays of arrays; Introduction to Structures and Enumerations

Procedures : Modular coding- Subroutines, Functions, Calling Functions and Subroutines; Arguments-Argument-passing mechanisms, Event-handler arguments, Passing an unknown no. of arguments, Named arguments, more types of function return values, overloading functions; Recursive functions;

Exception Handling and Debugging : Types of Errors-Design-time, Runtime and Logic errors; Exceptions and Structured Exception handling- studying an exception, getting a handle on this exception, finally, customizing exception handling, throwing your own exceptions; Debugging- Breakpoints, stepping through , local and watch windows.

Unit III: Object Oriented Programming & Custom Controls **16+08 Hours**

*Object Oriented Programming :*OOP Fundamentals – Class and objects, Creating Classes, Namespaces , Creating Property procedures, Class Methods, Class Constructors, Shared Methods, Shared Variables, Firing Events, Class Access Options, Inheritance, Polymorphism, Parent Class keywords, Derived class keywords, Parent class member keywords, Derived class member keyword, MyBase and MyClass.

Unit IV: MDI Applications, Library Functions & Files **08+04 Hours**

MDI Applications: MDI Applications Basics-Building an MDI application, Creating and accessing Child Forms, Ending an MDI application

Library Functions : Char Class – Properties : MaxValue, MinValue; Methods – IsDigit(), IsControl(), IsLetter(), IsLetterOrDigit(), IsLower(), IsUpper(), IsNumber(), IsWhiteSpace(), ToLower(), ToUpper(), ToString();
String Class – Properties : Length,Chars; Methods - Compare(), Concat(), Copy(), EndsWith(), StartsWith(), IndexOf(), LastIndexOf(), Insert(), PadLeft(), PadRight(), Remove(), Replace(), Join(), Split(), SubString(), ToLower(), ToUpper(), Trim(), TrimEnd(), TrimStart()
Math Class – Methods : Abs(), Ceiling(), Floor(), Log(), Max(), Min(), Pow(), Round(), Sign();
DateTime Class – Properties : Date, DayOfWeek, DayOfYear, Hour, Minute, Second, Day, Month, Year, TimeOfDay; Methods : Compare(), DaysInMonth(), IsLeapYear(), Add(), Subtract(), AddDays(), AddHours(), AddMinutes(), AddMonths(), AddSeconds(), AddYears(), Today(), ToLongDateString(), ToLongTimeString(), ToShortDateString(), ToShortTimeString(), ToString();

Unit V: Database and Web Programming **16+08 Hours**

ADO.NET: ADO.NET architecture, Creating a Dataset, DataGrid control, Data Binding, BindingContext object, Binding complex controls, DataAdapter object, Command object, DataReader object,

ASP.NET : Building a web application, maintaining state, web controls, validation controls, ASP.NET objects – Page object, Response object, Request object, Server object; Handling multiple forms in Web applications;

Database Access in Web Applications: Data-Bound Web controls – DropDownList DataList, DatGrid, Repeater; Simple Data binding, Binding to DataSets;

Instructions for Paper Setter

The question papers will be set according to the following scheme.

Theory				Practical		
Unit	**To be set**	**To be answered**	**Marks**	**To be set**	**To be answered**	**Marks**
I	2	1	08	2	1	11.5
II	2	1	08			
III	2	1	08	2	1	12
IV	2	1	05			
V	2	1	8.5	2	1	14
Total	**10**	**5**	**37.5**	**6**	**3**	**37.5**

Recommended Books

***Textbook*:**

1. **Anne Boehm,** *Murach's Visual Basic 2010*, First Edition, Shroff Publishers and Distributors Pvt. Ltd.

Reference books:

2. **Paul Deitel, Harvey Deitel, Abbey Deitel,** *Visual Basic 2012 How to Program (6th Edition)*, Prentice Hall, 2013
3. **Petroutsos, E.**, *Mastering Visual Basic.NET*, New Delhi: BPB Publications, 2004
4. **Jeffrey R Shapiro**, *Complete Reference Visual Basic.NET*
5. **Riordan**, *Microsoft ADO.NET Step By Step*, New Delhi: Prentice-Hall India, 2004
6. **Duthie**, *Microsoft ASP.NET Step By Step*, New Delhi: Prentice-Hall India, 2004

Practical Assignments

(Questions need not be restricted to this list)

Unit I

1. Design a form and place a TextBox in it. Call it (assign its name property) txtInput. Place a Command Button and call it cmdExtract. Assign the caption property of the Command button as "Extract". Write a program to extract each digit or letter of a number, word or sentence that is entered in txtInput and display them in a second Text Box called txtOutput one at a time on the click of a button.
2. A frmEmployee contains a TextBox (txtNumber) to enter number of employee records to be entered and two Command Buttons (cmdOK with the caption Ok and cmdClose with caption Close). As soon as a single digit number is entered, appropriate number of controls must be available in the form for entering Name, Address, Salary for the given number of employees. Write the code in appropriate Event to accomplish these.
3. Write a program to verify whether a given date is a valid date or not. Do not use library functions.
4. Design a form with suitable controls to input a single digit number and write appropriate event handlers to check if the number is automorphic or not. A number is called automorphic if the last digit of the square of the number is same as the number itself.(e.g., 6)
5. Design a form with suitable controls and write appropriate event handlers to list out all the Armstrong numbers within a given range of numbers 'm' to 'n'. A number is called an Armstrong number if the number is equal to the sum of the cubes of the digits of the number.

6. Design a form with suitable controls and write appropriate event handlers to generate an Ordinary Calculator Program (Using Label, CommandButton). The calculator should support the facilities such as Addition, Subtraction, Multiplication, Division, Storing in Memory, Clearing Memory and Adding to Memory etc. The display of the calculator should support up to 10 digits including decimal point. Your application should use control arrays
7. In the color code that is used in resistors, the different colors have values as follows: Black=0, Brown=1, Red=2, Orange=3, Yellow=4, Green=5, Blue=6, Violet=7, Gray=8 and White=9. The value of the resistor is indicated by drawing three colored bands round it. The first two bands indicate the first two digits in the numerical value of the resistance, while the third band is the decimal multiplier, i.e., it gives the number of zeros after the two digits. For example, if the bands have colors, Green-Blue-Orange, successively, then the numerical value is 56000. Design a form with suitable controls and write appropriate event handlers to accept the colors from the user and print the equivalent numerical.
8. Using functions, write a program to calculate the simple interest accrued on a given principal using the formula SI = (Principal x Rate x Time)/100. The user input and the output thereof must be on different forms. The input form must have a textbox where the principal will be entered by the user, a vertical scroll bar for the rate of interest, and a listbox from where the user can select the time (in years.) On clicking a button, the function must calculate the SI taking values from the textbox, scrollbar, and listbox, and the result shown in the second form. Provision must also be kept for adding and removing items to and from the listbox. The items in the listbox appear as: 1 year; 2 years; 3 years etc….. up to 10 years.
9. Write a program to calculate and display the factorial of a given number, using a recursive function.
10. Design a form with suitable controls and write appropriate event handlers to convert an input decimal number to a number with a user defined base (1 to 9), and vice versa.
11. Develop an application providing the facilities for a stopwatch, a timer, and a daily alarm at a preset time, as desired by the user. The selection of the option should be through radio buttons.
12. Load a picture on an appropriate control such that the position of the picture randomly changes within the form with time.
13. Design a form with suitable controls and write appropriate event handlers to load all the Colours (Using VScrollBox, HScrollBox).
14. Develop an application where all possible colours can displayed in a picture box using the three primary colours red, green, blue, whose values are selected from three scrollbars.
15. Load a picture on an appropriate control such that the position of the picture randomly changes within the form with time.
16. Write a program to sort the elements of an array in descending order using bubble sort, selection sort and insertion sort.
17. Write a program to search for an element in an array using binary search and linear search.

Unit II

18. Design a form with suitable controls and write appropriate event handlers to take in a string and determine whether the given string is palindrome or not.
19. Design a form with suitable controls and write appropriate event handlers to generate the calendar of a given month. The user must enter the month and the year. Assume that 1st January 1900 was a Monday.
20. Write a program to search for a particular word or pattern in a text and to display the position of the match. The match should also be selected.

21. Write a program to convert a string to proper case.
22. Write a program to input a string and perform the following tasks without using library functions: (a) to find its length, (b) to change it to upper case / lower case (c) to extract the left most n characters, (d) to extract the right most n characters (e) to extract n characters from it starting from position p, (f) to insert another string in it at position p (g) to replace n characters in it starting at position p with a given string
23. Write a program to analyze a line of text i.e. to count no. of words, digits, letters, special characters, vowels, consonants, no. of times a particular word appears.
24. Design a project that will enable you to simulate the Windows Explorer utility. Use Listview and Treeview controls.
25. Write a line of text and place it centered on a form. Ensure that the text remains centered even if the form is resized manually or otherwise.
26. A line of text E.g." Over to Delhi for the second day's play." is entered by the user. Write a program to print the shortest and longest word so contained in the sentence.
27. Develop a program to get the total file count and total size in a directory.
28. Create a class rectangle. The class has attributes length and width, each which defaults to 1. It has methods that calculate the perimeter and the area of the rectangle. It has set and get methods for both length and width. The set method should verify that length and width are each floating point numbers larger than 0.0 and less than 20.0. Create a class to test this Rectangle class.
29. Create a more sophisticated Rectangle class which stores only the Cartesian coordinates of the four corners of the rectangle. The constructor calls a set method that accepts four sets of coordinates and verifies that each of these are in the first quadrant with no single x or y coordinate more than 20.0. The set method also verifies that the supplied coordinates specify a rectangle. Provide methods to calculate the length, width, perimeter and area. The length must be the larger of the two dimensions. Include a Boolean method isSquare which determines if the rectangle is a square. Write a program to test this class.
30. Create a class savingAccount. Use static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest to calculate the monthly interest by multiplying the savingBalance by annualInterestRate divided by 12- this interest must be added to savingBalance. Provide a static method modifyInterestRate that sets the annualInterestRate to a new value. Write a program to test this class.
31. Create a class called complex for performing arithmetic with complex numbers of the form – real + imaginary * i, where i = $\sqrt{-1}$

    Write a program to test your class. Use floating point variables to represent the private date of the class. Provide a constructor that enables an object of this class to be initialized when created. Provide a no- argument constructor with default values in case no initializers are provided. Provide public methods that perform the following operations.

    i. Add two complex numbers. (a+bi) + (c+di)=(a+c) + (b +d) i
    ii. Subtract two complex numbers. (a+bi) - (c+di)=(a-c) + (b -d) i
    iii. Multiple complex numbers (a+bi) * (c+di)=(a*c –b*d) + (a*d+b*c) i
    iv. Print complex numbers.
32. An educational institution wishes to maintain a database of its employees. A staff member has a code and name. a staff member can be either a teacher, with a subject, or a typist, with typing speed, or an officer, with a grade. A typist can be a regular typist with a basic salary or a casual employee with

daily wage. Write a program to define all these classes and also define methods to store and retrieve values for these classes.

33. Assume that a bank maintains two kinds of accounts for its customers, one called saving account and the other current account. The saving account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Curr-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks –
    v. Accept deposit from a customer and update his balance.
    vi. Display the balance
    vii. Compute and deposit interest
    viii. Permit withdrawal and update balance
    ix. Check for the minimum balance, impose penalty, if necessary, and update balance.

Unit III

34. The following information is to be maintained regarding the users of electricity: Name, code and units consumed. Write a program that will take the name and units consumed, and hence generate a bill. For the first 20 units cost is 30p/unit, for the next 20 units, 40p/unit, for the rest, 50p/unit. Make provisions for reading, editing and deleting data. Make provisions to keep the rates alterable.
35. A publishing company maintains records with the following information: Name of Author, Author Code, Name of Book, Book ID and Year of Publication. Make provisions to add, edit and delete records. Every time a new Author Name is added, the code must be generated, so also with the name of book and book ID.
36. Refer to the above question to design a Crystal Report to display the details of the books for a given Author and given Year of Publication. Design a VB.NET form and write appropriate code to invoke the report.
37. Create a Database in a Database Server with two tables –Biodata and Marks. The table Biodata contains the fields Name, RollNo(N, 5)(RollNo is unique), Gender(C,1), State(C,15), District(C,15), Place(C,15), Class(C,3),Dob(Date), Caste(C,10) and the table Marks contains the fields RollNo(N,5), Physics(N,2), Chemistry(N,2), Maths(N,2). Develop ASP page(s) to Add, Edit and Delete records from the table. Provision should also be made to display all the records of a given class, along with each ones‟ average mark, in a tabular format (the class can be selected from a listbox).
38. Develop a web page that will calculate the monthly installment for a loan amount, given the rate of interest and its term.
39. Develop a web page with a counter that displays the number of visits to the site.
40. Develop a program to let user place order for ice cream over the net. This should allow selection of one or more flavours (vanilla, strawberry, etc.) and then select the item (cone, double cone, cups, etc.). The order summary should be displayed on the page once the user clicks on the *Order* button.
41. Develop a web site for a commercial organisation, where the order for goods can be placed. There should be possibility for adding new items or removing items from the shopping cart.
42. Develop a web site for registering the details of alumni for an educational institution. Make provisions for listing out the entries belonging to a particular batch.

CS502CT and CS502CP: Object Oriented Programming through Java

Objective:
The course is designed to impart knowledge and skill required to solve real world problems using object-oriented approach utilizing Java language constructs. This course covers the two main parts of Java i.e. Java Language and Java Library

After completion of the course students are expected to understand the following-

- Java tokens for creating expressions and creating data types.
- The way various expressions and data types are assembled in packages.
- Implementation of Inheritance, Exception handling and Multithreading in Java.
- Java I/O basics and Applets.
- Network Programming in Java.
- Accessing relational databases from Java program.
- Java Servlets.

Outline of the Course

Minimum Class Hours			Exam time (Hours)		Marks				
					Theory		Practical		
Theory	Practical	Total	Theory	Practical	External	Internal	External	Internal	Total
60	40	100	2	3	37.5	12.5	37.5	12.5	100

Unit	Topic	Minimum Class Hours			Marks Theory
		Theory	Practical	Total	
I	Introduction to Java Programming, Classes and Methods	12	08	20	7.5
II	Inheritance, Exception handling, Multithreading Enumerations and Autoboxing	12	08	20	7.5
III	Generics, Lambda expressions, String handling, java.lang, java.util and Input/Output	12	08	20	7.5
IV	Applet class, Handling Events and GUI programming with JavaFX	12	08	20	7.5
V	Networking, JDBC, Java Servlets	12	08	20	7.5
Total		**60**	**40**	**100**	**37.5**

Detailed Syllabus

Unit I: Introduction to Java Programming, Classes and Methods 12+08 Hours
Introduction to Java: Genesis and Overview, Java & Internet, Object-Oriented Programming features (Abstraction, Encapsulation, Inheritance and Polymorphism); Difference between (Java Script and Java, Java and C++, Java applet and Application), Java Development Kit (JDK) Java Virtual Machine (JVM), The Bytecodes, Compile & run a simple program
Constant, Variable, Data types & Arrays: Java Token & Keywords, Primitive Types, Integer literal, Floating point literal, Character literal, Boolean literal, String literal, declaring a variable,

Dynamic initialization, The scope and lifetime of variable, Type conversion and casting, Automatic type promotion in expression, Arrays (One-dimension, Multidimension), Alternative array declaration syntax

Operators: Arithmetic operators, Bitwise operators, Relational operators, Boolean logical operators, The assignment operator, Conditional operator, Operator precedence

Control statement: Decision making and Branching (*if*, Nested *if, if-else-if* ladder, *switch*, Nested *switch***,** The?**:** operator), Decision making and Looping (*while, do-while, for)* , Jump (*break, continue* and *return*)

Introduction to classes, methods and objects: The general form of a class, declaring objects, Assigning object reference variable, Introducing methods (Adding methods to a class, returning a value, Adding methods that takes parameters), Constructors, Parameterized constructor, The *this* keyword, Instance variable hiding, Garbage collection, the finalize() method, A stack class-an example, Overloading(methods, constructors), Using object as parameters, Argument passing, Returning objects, Recursion, Introducing Access control (public, private and protected), *static***,** *final*, nested and inner classes, String class, Command-line argument, Variable-Length arguments, Scanner (Constructors, Basics, setting delimiters)

Unit II: Inheritance, Exception handling, Multithreading, Enumerations and Autoboxing 12+08 Hours

Inheritance: Extending a class, Basics of Inheritance, Member access and inheritance, using **super**, creating a multilevel hierarchy, when constructors are called, method overriding, dynamic method dispatch, using abstract classes, using *final* with inheritance, the Object class

Packages and Interface: Packages (Defining a package, Finding packages and classpath, Access protection and importing packages), Interfaces (Defining, implementing and Applying Interfaces, Variables in interface, Interfaces can be extended)

Exception handling: Exception handling fundamentals, Exception types (uncaught exceptions, using *try* and *catch*) Nested try statement, multiple *catch* clauses, throw, *throws* and *finally***,** Java's built-in exceptions, user defined throwable, user defined exception subclasses, using Exception

Multithreaded Programming: The Java thread model (thread priorities, synchronization and inter-thread communication); The main thread, Creating a thread. isAlive( ), join( ), Deadlock, suspending, resuming and stopping

Enumerations, Type Wrappers and Autoboxing: Enumeration fundamentals, Java Enumerations are Class Types, Enumerations Inherit Enum, Type Wrappers, autoboxing and methods, autoboxing/unboxing occurs in expressions, autoboxing/unboxing Boolean and Character values

Unit III: Generics, String handling, java.lang, java.util and Input/Output 12+08 Hours

Generics: Introduction, Generic class with Two Type Parameters, General form of a generic class, Bounded Types, Using wildcard arguments, Creating a generic method, Generic Interfaces, Ambiguity errors, Generic Restrictions

java.util package – The Collection Framework – Collection Overview, Collection Interfaces (*Collection, List, Set, SortedSet),* Accessing Collection via an Iterator, Arrays

String handling: The string constructor, Special string operations, Character extraction, String searching & comparison, Data conversion using valueOf (), StringBuffer

*Exploring java.lang***:** Math functions (transcendental, exponential, rounding)

Input/Output-: Streams (Byte Streams and Character streams- class InputStream, OutputStream, Reader, Writer, Predefined streams, InputStreamReader, BufferedReader, Reading console input, writing console output, Reading and writing files, File, FileNameFilter & Directories, FileInputStream, FileOutputStream, PrintStream, FileReader, FileWriter

Unit IV: Applet Class and Handling Events 12+08 Hours

The applet class: Applet fundamentals, The applet class, Applet architecture, Applet skeleton (initialization and termination, overriding update()),Applet Display Methods, Requesting repainting, Using the Status Window, HTML applet tag, Passing parameters to applets

Handling events:, The Delegation Event Model, Event Classes, Sources of events, Event Listener Interfaces, Processing mouse events, Handling keyboard events, Adapter classes, Anonymous Inner classes

Unit V: Networking, JDBC, Java Servlets 12+08 Hours

Networking: Networking basics, InetAddress, Factory methods, URL, URLConnection, HttpURLConnection, Establishing a simple server using Stream Sockets, Establishing a simple client using Stream Sockets, Connectionless Client/Server Interaction with Datagrams

Java database connectivity (JDBC): Introduction to JDBC, type of JDBC connectivity, Accessing relational database from Java programs, Establishing database connections

Java Servlets: Background, The life cycle of a Servlet, The javax.servlet.http package (HttpServletRequest, HttpServletResponse, HttpSession Interfaces, Cookie class, HttpServlet class, HttpSessionEvent class), Handling Http requests and Response (HTTP GET & HTTP POST), Using Cookies, Session Tracking

Recommended Book

1. **Herbert Schildt**, *Java- The Complete Reference*, McGraw Hill Education (India) Ninth Edition, 2014.

Reference books*:*

1. **Deitel Deitel**, *Java How to program*, Prentice Hall India Ltd Ninth Edition, 2012.
2. **Kathy Sierra & Bert Bates**, *Head First Java*, O'Reilly Second Edition.
3. **E Balagurusamy**, *Programming with Java A Primer*, McGraw Hill Education Fourth Edition 2006.

Instructions to Paper Setter

Questions should be set according to the following scheme.

Theory Questions				**Practical Questions**		
Unit	**To be set**	**To be Answered**	**Marks**	**To be set**	**To be Answered**	**Marks**
I	2	1	09	2	1	13.5
II	2	1	09			
III	2	1	09	3	2	24
IV	2	1	09			
V	2	1	09			
Total	**10**	**5**	**45**	**5**	**3**	**37.5**

Practical Assignments

(Questions need not be restricted to this list)

1. Write a program to create a class called Box with a parameterized constructor, along with a method to calculate the volume of the box. Use the class to find the volume of two boxes whose height, width and depth are 10, 20, 30 and 20, 30, 40 respectively.
2. Define a class called stack that can hold 10 integer values, then initialize top of the stack, with push and pop methods. Write a program to push the elements into the stack and pop out from the stack.
3. Write a Java program using a class to multiply two matrixes of 3*3 order. Allow the user to input the values through the keyboard.
4. Write a program to multiply two numbers using a method in a class and pass the values using call by value (pass by value and pass by reference) techniques.
5. Write a Java program to find the factorial of positive integer using recursion.
6. Write a Java program to accept command line arguments and display the arguments along with the positions.
7. Write a Java program to demonstrate method overriding where the program creates a superclass called figure that stores the dimensions of various two-dimensional objects. It also defines a method called area() that computes the area of an object. The program derives two subclasses from figure. The first is Rectangle and the second is Triangle. Each of these subclass overrides area() so that it returns the area of a rectangle and a triangle respectively.
8. Write a Java program to create a thread and start running it using runnable interface. Allow the thread to display a message five times with a gap of 500ms.
9. Write a Java program to demonstrate the synchronization of two threads using the synchronized statement.
10. Write a Java program to demonstrate interthread communication considering the producer and consumer problem. There must be two classes one for producer to produce data and another is consumer to consume data [Hint: Use wait() and nothing() to signal in both directions].
11. Write a Java program to copy the contents of one file to another using java.io
12. Write an applet program to accept a message from the keyboard and then to display it on the console.
13. Write an applet to find the biggest of three numbers from the keyboard and display it on the console.
14. Design a Calculator System using Java, The applet should have all the digit buttons along with buttons for operations +,-,*,/ and =. There is a designated panel to show the current results. If a digital button is clicked, the number is displayed on the panel. If an operator button is clicked the operation is to be performed. You may assume the expression to be infix. The calculator can operate in two modes

i. When the operator buttons are pressed the intermediate results should be displayed

ii. The operations can take in any number of arguments and the final result is displayed only when the = button is pressed. [Hint: Use Overloading]

15. Write a program to input integers into an array and sort them using methods. Display the sorted numbers.
16. Write a generic method **printArray** that can print the string representations of the elements of Integer, Double and Character array. [Hint. use **public static < E> void printArray(E[] inputArray)**].
17. Write a generic class **Stack** with one type parameter say T. Provide the necessary constructor and the push() and pop() methods. Test this class with the primitive data types i.e a stack of integer types, stack of double types and stack of character types.
18. Write a program to copy the contents of one file to another file using command line arguments. Give appropriate error messages if any I/O error occurs.
19. Write a socket based Java application program to create a connection between two machines such that whatever text one machine is sending to the other will be displayed at the latter's screen and vice-versa
20. Create a Java application in which a particular machine is configured as the time server which continually listens for requests for time from clients. Clients request the server for time as a result of which the server sends the current time of the clients. The clients make a correction of the received time by adding a very small positive constant to the value and display the corrected time.
21. Create an editor applet in Java using which the users can enter some text and set the font and color of the text according to their choice. The text will be displayed appropriately when the applet is run.
22. Develop various programs that demonstrate the use of the mentioned JavaFX controls.
23. Develop a servlet allowing you to read the names and values of parameters that are included in a client request using ServletRequest class. Develop the web page corresponding to the servlet.
24. Develop a servlet that handles an HTTP GET request. The servlet is involved when a form on a web page is submitted. The HTML web page defines a form that contains a select elements and a submit buttons. The select element name is color and the options are Red, Green and Blue. The servlet responses according to the option submitted and display the message "you have selected color".
25. Develop a servlet that handles an HTTP POST request. The servlet is involved when a form on a web page is submitted. [Hint: The HTML source code is same as the above problem. Except that the method parameter for the form tag explicitly specifies that the POST method should be used and the action parameter for the form tag specifies a different servlet].
26. Write a Java program that prints the addresses and names of the local machine and two well known explored Internet web sites.
27. Implement a simple networked communications client and server. Messages are typed into the window at the server and written across the network to the client side and then they are displayed to demonstrate datagrams.

CS601T: Software Engineering

Objective: Software Engineering is a fast developing field. We can view Software Engineering as the engineering approach to developing software. The objective of this paper is to provide a broad understanding of system development concepts. It provides the students with a sense of confidence to develop new systems.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Marks		
		External	**Internal**	**Total**
50	2	37.5	12.5	50

Unit	Topic	Minimum Class Hours	Marks
I	System Analysis and Design	10	7.5
II	Introduction To Software Engineering Software life cycle models	06	6.5
III	Software project management, requirements and design	14	8.5
IV	Function Oriented Software Design, and User Interface Design	12	7.5
V	Testing, software reliability and maintenance	08	7.5
TOTAL		**50**	**37.5**

UNIT I **10 hours**

Introduction: System definition and concepts: Characteristics and types of system, Manual and automated systems

Systems models: Types of models, Systems environment and boundaries

Systems analyst: Role and need of systems analyst ,Qualifications and responsibilities, Change agent, Investigator and monitor, Architect, Psychologist, Salesperson, Motivator, politician, The analyst /User interface-behavioural issues, conflict resolution, MIS organization

System Planning: Data and fact gathering techniques: Interviews, Group communication, Presentations, Site visits.

***Feasibility study and its importance*:** Types of feasibility reports, System, Selection plan and proposal

UNIT II **6 hours**

Introduction: Evolution of an art to an engineering disciple, Solution to the software crisis, Computer systems engineering.

***Software life cycle models*:** Importance of life cycle model, waterfall model (feasibility study, requirement analysis and specification, design, coding and unit testing, integration and system testing, maintenance), prototyping model, evolutionary model, spiral model, Comparison of different life cycle models.

UNIT III **14 hours**

Software project management: Responsibilities of a Software Project Manager, Project Planning, Project Estimation Techniques (only Basic COCOMO), Scheduling (work breakdown, Activity Networks and Critical Path Method, Gantt Charts, PERT Charts, Project Monitoring and Control), Organization and

Team Structures (Organization Structure, Team Structure), Risk management (Definition), Software Configuration Management (Definition).

***Requirement Analysis and specification*:** Requirement gathering and analysis, Software Requirements Specification (Content of the SRS document, characteristics of a good SRS document, techniques for representing complex logic – Decision Tree and Decision Table).

Software Design: Characteristics of a good software design, cohesion and coupling (classification of cohesiveness and coupling), Software designs approaches (function-oriented design, Object-oriented Design).

UNIT IV **12 hours**

Function Oriented Software Design: Overview of SA/SD methodology, Structured Analysis, data Flow Diagrams (DFDs)(primitive symbols used for constructing DFDs, important concepts associated with designing DFDs, developing the DFD Model of a system, Shortcomings of the DFD Model), Structured Design (flow chart vs. structure chart, transformation of a DFD model into a structure chart).

Object Modelling using Design: Overview of Object-Oriented Concepts, Unified Modelling Language (UML), UML diagrams, USE CASE Model, Class Diagrams, Interaction Diagrams, Activity diagrams, State Chart Diagrams

User Interface Design: Characteristics of a good user interface, basic concepts (user guidance and online help, mode-based vs. Modeless Interface, Graphical User Interface (GUI) vs. Text-based User Interface), Types of user interfaces (command language-based Interface, Menu-based Interface, direct manipulation Interface), Component-Based GUI Development (Window system, Types of widgets, Visual programming)

UNIT V **8 hours**

Coding and Testing: coding standards and guidelines, code review (code walkthroughs, code inspection), Software Documentation(Internal and External), Testing (testing, verification vs. validation, design of test cases), Testing in the large, Testing in the small, unit testing, Black-box testing, White-box testing (Statement, Branch and Condition coverage), debugging, integration testing, system testing.

Software Reliability: Software reliability, software quality, and software quality management.

Software Maintenance: Characteristics of software maintenance (types of software maintenance, special problems associated with software maintenance), software reverse engineering.

Instruction to Paper Setter

Unit	Questions		Marks
	To be Set	To be Answered	
I	2	1	7.5
II	2	1	6.5
III	2	1	8.5
IV	2	1	7.5
V	2	1	7.5
Total	**10**	**5**	**37.5**

***Textbooks**:*

1. Elias M. Awad, ***System Analysis and Design***, Galgotia Publications (P) Ltd, New Delhi

2. Rajib Mall, *Fundamentals of Software Engineering,* Pearson Education/Prentice Hall of India, New Delhi.

References:

1. Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli, ***Fundamentals Of Software Engineering***, Second Edition, Prentice Hall of India Private Limited, New Delhi, 2002.

2. Richard E Fairley, ***Software Engineering Concepts***, Tata McGraw Hill Publishing Company Limited, New Delhi, 1997.

CS601P: Project

Objective

The objective of the project is to consolidate the concepts and practices that were learned during the course and to serve as a record of competence. It should enable a student to apply concretely in a small package the concepts gained from Software Engineering.

Outline of the Course

Minimum Hours	Marks		
	External	Internal	Total
75	37.5	12.5	50

Guidelines

- ***Overview*:** The project will be carried out over a duration of three months, involving minimum 100 hours for General students and minimum 180 hours for Honours students. Every student should do a project individually and not in a group. The selected project can be either of type Model 1 or Model 2 described below.
- ***Platform*:** The project can be in any platform e.g., DOS, WINDOWS, UNIX, LINUX, Mac OS, etc.
- ***Language and package*:** The project can be done using any language or package learned within or outside the course such as C, C++, Java, VB, C#, Director, tcl, VC++, Visual FoxPro, Flash, etc.
- ***Venue*:** The project can be done in the College itself or in a reputed organization.
- ***Guides*:** Internal Guides from within the college should be assigned to each student. If the project is to be done in a reputed organization, an External Guide from that organization is also required as Co-Guide, and the qualification of the External Guide should not be less than that of the Internal Guide.
- ***Monitoring of Projects*:** The progress of the project should be monitored through seminars, and each of the seminars should be evaluated, a record of which should be maintained. Every student will have to maintain a log book where the coding of the project is kept. This will have to be periodically signed by the internal guide. The number of seminars should not be less than three (e.g. Analysis, Design, Implementation).
- ***Final Examination*:** For the final external evaluation a brief summary of the project should be submitted to the university at least one week prior to the date of the examination for the benefit of the external examiner(s).

Types of Project

Model 1

1. The topic for the project can be any subsystem of a system software or tool or any scientific or a fairly complex algorithmic situation.
2. The aim of this type is to highlight the abilities of algorithmic formulation, program and data flow representation, modular programming or object oriented programming, optimized code preparation, systematic documentation and other associated aspects of software engineering.
3. The assessment would be through the Project Report, Viva and the following criteria for this model:
   - Programming style, structured design, minimum coupling and high cohesion, abstraction, encapsulation, inheritance and polymorphism, as relevant.
   - Good commenting and annotating of the code and flow of representation, such that meaningful code, with good readability and ease of maintenance, results.
   - Design specifications, depicting the method adopted and giving a simple data dictionary for each data, to cover name, type and validity aspects.

» Test case samples, enough in number, to adequately cover the possible chances of common errors

Model 2

1. This model can be of a typical business application. The aim of this type is to highlight the stages involved in a typical business oriented project development, though on a miniature scale, in a real or simulated environment. The appropriate use of DBMS/RDBMS towards any business application, along with adequate system analysis and structured design and development of specific tools/products, would be the underlying activity in preparing this project.
2. The emphasis should be on selecting a system/subsystem that shows the DBMS/ RDBMS and System Analysis aspects to a greater degree. Any small and simple business system may be selected, although candidates are advised to use their knowledge and creativity, to select typical and intelligent applications, rather than run-of-the-mill themes, such as simple Pay roll calculation or Issue-Return portion of an inventory scheme. The Evaluation stage would give due weightage for theme selection, problem analysis, fact finding techniques and initial design, which is as close to real-life business situations as possible.
3. The code can be generated out of 4 GL Interface, like Screen Builder and Report Generator, Application Generator/Program Code Generators, or can be totally hand-coded or a combination of both. The documentation need not contain the code generated by these applications, but only that written by the candidate.
4. The assessment would be through the Project Report, Viva and the following criteria for this model:
   - » Requirements leading to the project, those which were the result of System Analysis
   - » The design aspect of DBMS/RDBMS oriented documentation which describes the structure and organization of the database, well annotated source code, supplemental documentation, which can serve as Data Analysis and Data Flow description
   - » A simple Data Dictionary of the elements which form the structure
   - » Details about I/O Screens and facilities for on-screen querying, print oriented Reports and built in house-keeping routines which help disk management and file integrity, are to be included to the extent possible.
   - » Details of Acceptance Tests which, should be in adequate number and should include error messages

Content of the Project Report

1. Acknowledgement
2. Certificate, stating it to be a bonafide work of the student, and that it has not been submitted for any other examination, and counter-signed by the project guide(s).
3. Synopsis of the project
4. Description of the existing system
5. Proposed system
6. User requirements
7. Hardware and software requirements
8. Costs and benefits estimation
9. Gantt Chart (Project Control)
10. System Flow Charts, Algorithms
11. DFD, Decision Tables, Decision Trees
12. Data Dictionary
13. Module Design
14. Database Design
15. File Description

16. Source Code
17. Input and Output Screen Design
18. Testing used and Test Results
19. Need for review : deficiencies and future enhancements
20. User/Operational Manual (including menu design, security aspects, access rights, backup, controls etc.)

Data Dictionary

1. This should give a catalogue of the data elements used in the system/subsystem developed.
2. The following are the details required. Write NA where NOT applicable
   - » Data Name
   - » Aliases, if any
   - » Length (size)
   - » Type (Numeric, alpha, binary, etc)
   - » Validity criterion (Minimum, maximum, etc)
   - » Default value, if any
   - » Whether related to other data items
   - » Where used in the program: Reference to data structure/file/procedures/modules

User Manual

It may include chapters like the ones suggested below:

- » Installation
- » Hardware requirements
- » System requirements
- » Installation procedure, including security aspects like password, protection, backups, controls, etc
- » Menu choices and their actions - screen formats
- » Error messages
- » Output
- » A Sample test case

Viva-Voce

The viva-voce will be conducted by external examiner(s) appointed by the University and internal examiner(s) from the College. Other members of the faculty and students may be present. It will be of duration of about 15 to 20 minutes. The analysis, design aspects and quality of implementation of the project would be the main subject matter for the viva. However the general proficiency of the candidate in the selected software platform should also be tested.

Distribution of Marks

Sl. No	Criteria	Marks
1.	Analysis	7.5
2.	Design	7.5
3.	Implementation	7.5
4.	Project Report	7.5
5.	Viva	7.5
6.	Internal Assessment	12.5
	Total	**50**

CS602AT: Compiler Design

Objective: The objective of the theory of computing is to introduce and study abstract, mathematical models of computation (such as finite state, push down & Turing machines), and to use the abstract machines models to study the ability to solve computational problems. One such application is a compiler. It is important to understand the process by which programs written in high-level languages are translated and executed. After completing the course, students will be able to use regular expression effectively and appropriately, construct derivations and parse trees, understand the equivalence of grammars, languages & automata and translate between grammars, languages & automata.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Marks		
		External	Internal	Total
100	3	**75**	**25**	100

Unit	Topics	Minimum Class Hours	Marks
I	Introduction to Compiler Techniques, Lexical Analysis	20	15
II	Syntax Analysis	20	15
III	Type Checking and Symbol Table	20	15
IV		20	15
V	Runtime Environment Management and Intermediate Code Generation	20	15
Total		**100**	**75**

Detailed Syllabus

Unit I: Introduction to Compiler Techniques, Lexical Analysis **20 hours**

Introduction- Phases of a compiler, compiler applications
Lexical Analysis-Role of the Lexical Analyzer, Input Buffering, Specification of Tokens, recognition of Tokens

Unit II: Syntax Analysis **20 hours**

Introduction to syntax analysis, Grammar, Top-Down parsing (Recursive Descent Parsing, Recursive Predictive Parsing, Non-recursive Predictive Parsing-LL(k) parsing), Bottom-Up parsing, LR parsing (SLR Parsing tables, LR (1) parsing, LALR(1) parsing)

Unit III: Type Checking and Symbol Table **20 Hours**

Type Checking – Static vs. Dynamic Checking, Type Expressions, Type Checking, Type equivalence, Type Conversion
Symbol Table **–** Information in symbol table, Features of symbol tables, simple symbol table, scoped symbol table

Unit IV: Runtime Environment Management & Intermediate Code Generation 20 Hours

Runtime Environment Management –Activation Record, Display
Intermediate Code Generation – Intermediate Languages, Intermediate language design issues, Intermediate representation techniques, statements in three-address code, three-address code generation (code generation for arrays, translation of Boolean expressions, translation of control flow statements)

Unit V: Target Code Generation and Code Optimizatin 20 Hours

Target Code Generation **–**Factors affecting code generation, basic block, code generation for trees, register allocation, code generation using dynamic programming
Code Optimization- Introduction and issues in optimization, optimizing transformations, local optimization, global optimization;

Instructions for Paper Setter

Unit	Questions	
	To be Set	**To be answered**
I	2	1
II	2	1
III	2	1
IV	2	1
V	2	1
	10	**5**

Recommended Books

Textbooks:

1. **Mishra, K. L. P.; N. Chandrasekaran,** *Theory of Computer Science* (Third Edition)**,** New Delhi: PHI Learning Private Limited, 2011
2. **Chattopadhyay, S.,** *Compiler Design,* PHI Learning Pvt. Ltd, 2009

Further Reading

1. **Martin, J. C.,** *Introduction to Languages and the Theory of Automata,* New Delhi: Tata McGraw-Hill
2. **Lewis, H. R.; C. H. Papadimitriou,** *Elements of the Theory of Computation* (Second Edition), New Delhi: Prentice-Hall India, 2003
3. **Hopcroft, H. E.; J. D. Ullman,** *Introduction to Automata Theory, Languages and Computation,* New Delhi: Narosa Publications, 2001
4. **Aho, A. V.; Lam, M. S.; Sethi, R.; Ullman, J. D.,** *Compilers principles, techniques and tools* (Second Edition), Pearson Education, Fifth Impression,2011
5. **Louden, K.,** *Compiler Construction: Principles and Practice,* Cengage Learning, 1997
6. **K. Muneeswaran,** *Compiler Design,* Oxford University Press, 2013
7. **Papadimitriou, C. H.,** *Computation Complexity***,** New Delhi: Addison Wesley

CS602BT : Artificial Intelligence

Objective: Artificial Intelligence has embraced the larger scientific goal of constructing an information-processing theory of intelligence. If such a science of intelligence could be developed, it could guide the esign of intelligent machines as well as explicate intelligent behavior as it occurs in humans and other animals. This paper describes the fundamental AI ideas that underlay many of the AI applications and provides a base for understanding natural intelligence.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Marks		
		External	Internal	Total
100	3	75	25	100

Unit	Topic	Minimum Class Hours	Marks
I	Overview and Problem Solving Techniques	20	15
II	Heuristic Search and Adversarial Search	20	15
III	Knowledge Representation using Logic and Reasoning	20	15
IV	Expert Systems and Introduction to Advanced AI	20	15
V	Programming in Prolog	20	15
Total		**100**	**75**

Detailed Syllabus

Unit I: Overview and Problem Solving Techniques 20 **Hours**
Definition of AI, foundations of AI, AI technique, State space, defining a problem as a state space, Uninformed search strategies - Breadth-First, Depth-First, Iterative deepening

Unit II: Heuristic Search and Adversarial Search **20 Hours**
Heuristic functions, Heuristic search strategies (Best-First Search, A*, Hill climbing search, Steepest Ascent Hill climbing), Problem Reduction search: AO* algorithm, Constraint satisfaction problems, Adversarial Search (Games, Minimax algorithm, alpha-beta pruning)

Unit II: Knowledge Representation using Logic and Reasoning **20 Hours**
Knowledge representation using logic, representing facts in logic, representing instances and ISA relationships, computable functions and predicates, resolution, conversion to clausal form, basis of resolution, resolution in propositional logic, unification, resolution in predicate logic, question answering, Representation of knowledge using rules: Procedural vs. Declarative Knowledge, Logic Programming, Forward vs. Backward Reasoning, Matching, Control Knowledge.

Unit IV: Expert Systems and Introduction to Advanced AI **20 Hours**
Expert System: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.
Natural Language Processing: Overview, Steps in the process: Morphological Analysis, Syntactic Analysis, Semantic Analysis, Discourse Integration, Pragmatic Analysis
Learning: Rote Learning, Learning by Taking Advice, Learning by Induction

Unit V: Programming in Prolog **20 Hours**

Introduction to Prolog, Syntax and meaning of Prolog programs, Lists, Operators, Arithmetic, Using structure, Controlling Backtracking, Input and Output, Built-in Predicates - , Basic Problem solving – depth first, breadth-first, iterative deepening, best-first, AND/OR graph representation, Knowledge representation and expert system – function and structure of an expert system, representing knowledge with if-then rules, forward and backward chaining in rule based system

Instructions for Paper Setter:

Unit	Questions	
	To be Set	**To be Answered**
I	2	1
II	2	1
III	2	1
IV	2	1
V	2	1
Total	**8**	**5**

Recommended Books

1. Rich, E.; Knight K.; Nair S. B. , Artificial Intelligence, (Third Edition), New Delhi: Tata McGraw-Hill, 2009
2. Ivan Bratko; ROLOG : Programming for Artificial Intelligence,3rd Edition, New Delhi: Pearson India, 2009

Further reading

1. Russell, S.; P. Norvig, Artificial Intelligence: A Modern Approach (Second Edition), Pearson Education Inc., 2003
2. Sivanandam, S.N. ; Deepa, S. N. : Principles of Soft Computing (Second Edition), New Delhi, Wiley India Pvt. Ltd., 2012
3. Negnevitsky, M., Artificial Intelligence: A Guide to Intelligent Systems (Second Edition), New Delhi: Pearson Education Inc., 2002
4. Giarratano J. C.; Riley G. D., , Expert Systems : Principles and Programming, New Delhi, Cengage Learning , 2005

CS602CT: Data Mining

Objective: Capabilities of both generating and collecting data have been increasing rapidly in the last several decades due to the use of bar codes, computerizations of many products, advances of data collection tools etc. This growth in stored data has generated an urgent need of the subject like Data Mining. The paper aims to give the concept and various techniques of data mining to the students. Students will also learn the feasibility, usefulness, efficiency and the scalability of the techniques for discovery of patterns hidden in large databases.

Outline of the course

Minimum Class hours	Exam Time	Marks		
		External	Internal	Total
100	3 hours	75	25	100

Unit	Topic	Class Hours	Marks
I	Introduction	07	5
II	Data Preprocessing and Data warehouse	20	15
III	Mining Frequent Patterns, Association and Correlations	30	20
IV	Classification, Prediction, Cluster Analysis and Outlier Analysis	35	30
V	Applications and Trends	08	5
Total		100	75

Detailed Syllabus

Unit 1: Introduction **07 hours**

What motivated data mining? Why is it important? What is data mining? Data mining-on what kind of data? Data mining functionalities. Are all patterns interesting? Classifications of data mining systems. Data mining task primitives. Major issues in data mining.

Unit II: **Data Preprocessing and Data Warehouse** **20 hours**

Why process the data? *Descriptive Data Summarization*: Mesuring central tendency, Measuring the dispersion, Garphic displays. *Data Cleaning*: Missing values, Noisy data, Data cleaning as a process. *Data Integration and Transformation*: Data integration, Data transformation. *Data Reduction*: Data cube aggregation, Attribute subset selection, Dimensionality reduction and Numerosity reduction. Data discretization and concept hierarchy generation.

***Data Warehouse*:** What is data warehouse? *A Multidimensional Data Model* : tables and spreadsheets to data cubes; stars, snowflakes and fact constellations; Measures, Concept hierarchies, OLAP operations in the multidimensional data model. *From Data Warehouse to Data mining*: Data warehouse usage, From online analytical processing to online analytical mining.

Unit III: Mining Frequent Patterns, Association and Correlations **30 hours**

Basic Concepts: Market basket analysis; Frequent itemset, Closed itemsets and association rules; Frequent pattern mining. *Efficient and Scalable Frequent Itemset Mining Methods*: The Apriori algorithm, Generation association rules from frequent itemsets, Improving the efficiency of Apriori, Mining frequent itemset without candidate generation, Mining frequent itemset using vertical data format,

Mining closed frequent itemsets. *Mining Various Kinds of Association Rules*: Mining multilevel association rules, Mining multidimensional association rules, *From Association Mining to Correlation Analysis*: Strong rules are not necessarily interesting, From association analysis to correlation analysis. *Constraint-based Association Rule Mining*: Metarule-guided mining of association rules, Constraint pushing.

Unit IV: Classification, Prediction, Cluster Analysis and Outlier Analysis 35 hours

Classification and Prediction: What is classification? What is prediction? *Classification by Decision Tree Induction*: Decision tree induction, Attribute selection measures, Tree pruning. *Prediction*: Linear regression, Nonlinear Regression.

Cluster Analysis: What is cluster analysis? *Types of Data*: Interval-scaled variables, Binary variables, Categorical, Ordinal, Ratio-scaled, Variables of mixed types and vector objects. Categorization of major clustering methods. *Partitioning Methods*: k-means and k-medoids, *Partitioning in Large Databases*: k-medoids to CLARANS. *Hierarchical Methods*: Agglomerative and divisive hierarchical clustering, BIRCH, ROCK. *Density-based Methods*: DBSCAN, OPTICS. *Grid-based methods*: STING. *Model-based methods*: Expectation-maximization, Conceptual clustering

Outlier analysis: Statistical distribution based outlier detection, Distance-based outlier detection, Density-based outlier detection, Deviation based outlier detection.

Unit V: Applications and Trends 08 hours

Data mining for financial data analysis, Data mining for retail industry, Data mining for telecommunication industry, Data mining biological data analysis, Data mining for scientific applications, Data mining for intrusion detection. *Data Mining for System Products*: How to choose a data mining system, Examples of commercial data mining systems. *Additional Themes*: Theoretical foundations, Statistical data mining, Visual and audio data mining, Data mining and collaborative filtering. Social impacts of data mining. Trends of data mining.

Instruction to Paper Setter

Unit	To be set	To be answered
I	2	1
II	2	1
III	2	1
IV	2	1
V	2	1
Total	**10**	**5**

Text Book

J Han and M Kamber. *Data Mining: Concepts and Techniques*, 2nd Edition. Morgan Kaufman Publisher/ Elsevier, 2010.

Reference books

1. **D Hand, H Mannila, P Smyth.** *Principles of Data Mining,* Prentice-Hall India, 2005
2. **A K Pujari.** *Data Mining Techniques,* University Press, 2008
3. **P Tan, V Kumar, M Steinbach.** *Introduction to Data Mining.* Pearson Education, 2007.