Syllabus for

BACHELOR OF SCIENCE
(COMPUTER SCIENCE)

NORTH EASTERN HILL UNIVERSITY
SHILLONG

Contents

Preamble

Overview

The Three Year B.Sc. Course in Computer Science, for both Pass and Honours students, builds on the recently revised course of Computer Science for classes VIII to XII. However, it is also a fact that many students have not had the opportunity to take up this subject at the pre-degree or they have studied it only in the non-formal sector. Hence, provision has been made for such situations by offering an elective in Paper I, which covers a syllabus similar to that of Class XII. For the long term it is envisaged that only students who have completed a course in computer science at the Higher Secondary stage would be permitted to take up this course at the Degree level.

Eligibility

Students who have satisfied the University norms if they had taken up Computer Science in Class XII, or who have knowledge of Programming in C and 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:

10% : Syntax and Input/Output screens

30% : Logic and efficiency (source code, pseudo code, algorithm)

20% : Error trapping (Illegal or invalid input, stack overflow, underflow, insufficient physical memory, etc.)

20% : Completion

20% : Result

COURSE OUTLINE (GENERAL & HONOURS)

Year	Stream	Paper	Name	Minimum Class Hours			Exam Time(Hours)		Marks			
				Theory	Practical	Total	Theory	Practical	Theory	Practical	Internal	Total
I	General	I	Data Structures Using C	60	60	120	2	3	50	40	10	100
	Honours	VI	Database Management System	95	25	120	3	-	90	-	10	100
II	General	II	Elective 1: Computer Graphics Elective 2: Visual Programming Using VB.NET	60	60	120	2	3	50	40	10	100
	General	III	Computer Organization and Architecture	120	-	120	3	-	90	-	10	100
	Honours	VII	Computer Networks	120	-	120	3	-	90	-	10	100
III	General	IV	Software Engineering	60	-	60	2	-	40	-	10	50
	General	V	Project	-	100	100	-	-	-	40	10	50
	Honours		Project	-	180	180	-	-	-	50	20	70
	Honours	VIII	Elective 1: Object Oriented Programming through Java Elective 2: Web Programming in PHP and MySQL	60	60	120	2	3	50	40	10	100
	Honours	IX	Operating System and Introduction to Linux	120	-	120	3	-	70	-	10	80

COURSE OUTLINE FOR GENERAL STUDENTS

Year	Stream	Paper	Name	Minimum Class Hours			Exam Time(Hours)		Marks			
				Theory	Practical	Total	Theory	Practical	Theory	Practical	Internal	Total
I	General	I	Data Structures Using C	60	60	120	2	3	50	40	10	100
II	General	II	Elective 1: Computer Graphics Elective 2: Visual Programming Using VB.NET	60	60	120	2	3	50	40	10	100
	General	III	Computer Organization and Architecture	120	-	120	3	-	90	-	10	100
III	General	IV	Software Engineering	60	-	60	2	-	40	-	10	50
	General	V	Project	-	100	100	-	-	-	40	10	50
Total Marks									**230**	**120**	**50**	**400**

COURSE OUTLINE FOR HONOURS STUDENTS

Year	Paper	Name	Minimum Class Hours			Exam Time(Hours)		Marks			
			Theory	Practical	Total	Theory	Practical	Theory	Practical	Internal	Total
I	I	Data Structures Using C	60	60	120	2	3	50	40	10	100
	VI	Database Management System	95	25	120	3	-	90	-	10	100
II	II	Elective 1: Computer Graphics Elective 2: Visual Programming Using VB.NET	60	60	120	2	3	50	40	10	100
	III	Computer Organization and Architecture	120	-	120	3	-	90	-	10	100
	VII	Computer Networks	120	-	120	3	-	90	-	10	100
III	IV	Software Engineering	60	-	60	2	-	40	-	10	50
		Project	-	180	180	-	-	-	50	20	70
	VIII	Elective 1: Object Oriented Programming through Java Elective 2: Web Programming in PHP and MySQL	60	60	120	2	3	50	40	10	100
	IX	Operating System and Introduction to Linux	120	-	120	3	-	70	-	10	80
Total								**530**	**170**	**100**	**800**

Paper I : Data Structures Using C

O bjective

The objective of the course is to present the students to learn how to create data structures in a computer language, such as C, to represent a collection of similar data, and how to process these data most efficiently for solving problems. 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 algorithms to solve real world problems
- » implement various data structures viz. Stacks, Queues, Linked Lists, Trees and Graphs
- » understand various searching and sorting techniques and their processing efficiency.

It is expected that the student has basic knowledge of C language.

Outline of the Course

Minimum Class Hours			Exam Time (Hours)		Marks			
Theory	Practical	Total	Theory	Practical	Theory	Practical	Internal	Total
60	60	120	2	3	50	40	10	100

Unit	Topic	Minimum Class Hours			Marks
		Theory	Practical	Total	(Theory)
I	Data Representation and Algorithm Design; Arrays	10	10	20	10
II	Linked Lists; Stacks and Queues	16	18	34	15
III	Trees and Graphs	18	18	36	15
IV	Searching and Sorting and their complexity analysis	15	15	30	10
TOTAL		60	60	120	50

Detailed Syllabus

Unit I : Data Representation and Algorithm Design; Arrays **10 Hours + 10 Hours**

Data Type, Abstract Data Type, Data Structure, Fundamental and Derived Data Types. Algorithm definition. Complexity measures in terms of time and space; Big O notation. Overview of I/O functions, Loops, Control Statements and functions in C. Recursion (Towers of Hanoi, Fibonacci Numbers, Binary search), Comparison of Recursive and Non-Recursive algorithms

Array as a data structure (characteristics, advantages, disadvantages), Representation of arrays: single and multidimensional, Address calculation using column and row major ordering; insertion and deletion in arrays; use of arrays for matrix representation and manipulation (addition, multiplication, transpose), complexity analysis for matrix multiplication. Use of arrays for large number representation.

Unit II : Linked Lists; Stacks and Queues **16 Hours + 18 Hours**

Defining and processing structures. Structure and arrays, Structure and Pointers. Self referential structures. Linked List as a data structure (characteristics, advantages, disadvantages); operations on lists (creation, insertion, deletion, traversal, merging, splitting); singly linked list (with one or two external pointers), doubly linked list, circular list; use of linked lists for polynomial representation and manipulation (addition and multiplication), and sparse matrix representation and manipulation (inputting, adding, and displaying in matrix form)

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

Unit III : Trees and Graphs **18 Hours + 18 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 arrays and linked lists, Binary tree traversal methods (pre-order, in-order, post-order), recursive and non-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), Definition and characteristics of threaded binary trees, multi-way search trees and B-tree.

Definition of a graph, Basic Terms (vertex, arc, directed, undirected, cardinality, finite and infinite graph, incidence, adjacency, indegree, outdegree, path length, weighted graph, connected graph, cyclic and acyclic graph, symmetric graph, complete graph, sub-graph); 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 IV : Searching and Sorting, and their complexity analysis **15 Hours + 15 Hours**

Linear and binary search, Indexed 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, Radix) and comparison of their time complexity.

Practical Assignments

(Questions need not be restricted to this list)

1. The *median* of an array of numbers is the element *m* of the array such that half the remaining numbers in the array are greater than or equal to *m* and half are less than or equal to *m*, if the

number of elements in the array is odd. If the number of elements is even, the median is the average of the two elements m_1 and m_2 such that half the remaining elements are greater than or equal to m_1 and m_2 and half the elements are less than or equal to m_1 and m_2. Write a C program to input numbers into an array and returns the median of the numbers in the array, using functions.

2. A complex valued matrix X is represented by a pair of matrices (A, B), where A contains real values and B contains imaginary part. Write a program to input values to these matrices, compute their product to give matrix Z, and display the three matrices X, Y, Z in matrix form. Determine the number of multiplications that will be done if X has m rows and n columns, and Y has p columns.

3. Write a program using structures to read the current date and print the next date.

4. Write a program to calculate the address of a given element in an array declared as float F[5][6][7][4]; and compare the result with that assigned by the compiler.

5. A vector is an array which can have its size increased as and when required. Elements can be deleted and inserted in sequential order. Write a program for implementation of vectors

6. 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

6. 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.

7. 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.

8. Write a program to implement polynomial multiplication. Test your program by inputting the following two polynomials given below.

   10 P^8 + 14 P^6 - 8 P^5 - 3 P^4 + P^2

   3 P^ 4 + 5 P^3 - 2 P + 9

   (^ is to be read as "raised to")

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.

9. 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.
10. 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.
11. 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

    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.
12. Write INSERT and DELETE functions in C language simulating insertion and deletion in circular queue which stores an array of characters.
13. 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.
14. 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.
15. 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.
16. 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.]
17. 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.
18. Write a program to create a binary tree and to traverse the tree in
    a) Recursive and non-recursive pre-order
    b) Recursive and non-recursive in-order

c) Recursive and non-recursive post-order

19. Implement a procedure for deleting an element *X* from a binary search tree.
20. Write a program to reconstruct a binary search tree given its pre-order and in-order traversal sequence.
21. Write a program to find the biggest and smallest item in a binary search tree.
22. Design and implement an algorithm for insertion of an element in AVL tree taking into account all possible conditions.
23. Represent a graph using adjacency matrix. Write a C procedure to transform an adjacency matrix based representation to a linked-list based representation.
24. Design a suitable representation so that a graph can be stored on a hard disk. Write a procedure adjacency matrix based representation.
25. Write a program to represent a graph and perform a non-recursive depth first search of an item in it.
26. Write a program to represent a graph and compute the shortest distance between two nodes in it.
27. Write a program to input some numbers into an array, and then sort them using various sorting techniques (selection sort, bubble sort, merge sort, quick sort, radix sort) and compare their time-complexities.
28. 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.
29. 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.

Instruction For Paper Setter

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

Unit	Theory Questions		Practical Questions		
	To be set	To be answered	To be set	To be answered	Marks
I	2	1	2	1	15
II	3	2			
III	3	2	2	1	15
IV	2	1	2	1	10

Distribution of marks for practical

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

Recommended Books

Text Book

1. **S. Chattopadhyay, D. Ghosh Dastidar, M Chattopdhyay**, *Data Structures Through C Language,* BPB Publications, 2001

Reference

1. **Y. Langsam, M.J. Augenstein, A.M. Tenenbaum,** *Data Structures Using C and C++*, Second Edition, Prentice Hall of India, 2000
2. **Seymour Lipschutz**, *Theory and Problems of Data Sturctures,* Schaum's Outline Series, International Edition, MacGraw Hill, 1986
3. **Niklaus Wirth**, *Algorithms + Data Structures = Programs,* Prentice Hall of India, 1998
4. **E. Horowitz, Sahni, D. Mehta**, *Fundamentals of Data Structures in C++,* Galgotia Publication, 2002
5. **Y.P. Kanetkar**, *Data Structures Through C Language,* BPB Publications, 2002

Paper II
(Elective 1) Computer Graphics

Objective

Computer graphics is one of the most exciting and rapidly growing computer fields. 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 computer graphics concepts and algorithms to implement the concepts.

Outline of the Course

Minimum Class Hours			Exam Time (Hours)		Marks			
Theory	Practical	Total	Theory	Practical	Theory	Practical	Internal	Total
80	40	120	2	3	50	40	10	100

Unit	Topic	Minimum Class Hours			Marks
		Theory	Practical	Total	Theory
I	Overview of Computer Graphics	10		10	5
II	Output Primitives	20	20	40	15
III	Two Dimensional Geometric Transformation	20	20	40	15
IV	Two Dimensional Viewing	20		20	10
V	Three Dimensional Concept	10		10	5
Total		80	40	120	50

Detailed Syllabus

Unit I **10 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 **20+20 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 **20+20 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.

Unit IV **20 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 **10 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.

Instructions for Paper Setter

<table>
<tr><th colspan="3">Theory Questions</th><th colspan="3">Practical Questions</th></tr>
<tr><th>Unit</th><th>To be set</th><th>To be Answered</th><th>To be set</th><th>To be Answered</th><th>Marks</th></tr>
<tr><td>I</td><td>2</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>II</td><td>2</td><td>1</td><td rowspan="2">2</td><td rowspan="2">1</td><td rowspan="2">40</td></tr>
<tr><td>III</td><td>2</td><td>1</td></tr>
<tr><td>IV</td><td>2</td><td>1</td><td rowspan="2"></td><td rowspan="2"></td><td rowspan="2"></td></tr>
<tr><td>V</td><td>2</td><td>1</td></tr>
</table>

Text Book

Hearn, D.; M. P. Baker, *Computer Graphics (*Second Edition)**,** New Delhi: Prentice-Hall India., 1994

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), Tata Mc-Graw Hill, 2002.

Paper II
Elective II: Visual 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	Total	Theory	Practical	Theory	Practical	Internal	Total
60	60	120	2	3	50	40	10	100

Unit	**Topic**	**Minimum Class Hours**			
		Theory	**Practical**	**Total**	**Marks (Theory)**
1	Language Fundamental, Forms and Controls	12	12	24	10
2	User-Defined Data Types, Procedures & Exception Handling	12	12	24	10
3	Object Oriented Programming & Custom Controls	12	12	24	10
4	MDI Applications, Library Functions & Files	12	12	24	10
5	Database and Web Programming	12	12	24	10
TOTAL		**60**	**60**	**120**	**50**

Detailed Syllabus

Unit I: Language Fundamentals, Forms & Controls **12 Hours + 12 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…Then…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 Vs 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;

Advanced Windows Controls - ListView, TreeView, DateTimePicker, Common Dialog Controls, DomainUpDown, NumericUpDown, RichTextBox, CrystalReportViewer;

Unit II: User-Defined Data Types, Procedures & Exception Handling **12 Hours + 12 Hours**

Arrays, Structures and Enumerations: Arrays-Declaring and Initializing arrays, array limits, multi-dimensional arrays, dynamic arrays, arrays of arrays; Creating and using Structures; Creating and using 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 **12 Hours + 12 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

Creating Custom Controls: Enhancing existing controls – Inheriting from Textbox, Listbox and Checkbox, Building Compound controls.

Unit IV: MDI Applications, Library Functions & Files **12 Hours + 12 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();

Working with Files and Folders: Accessing Files and Folders – Directory Class, File Class, DirectoryInfo Class, FileInfo Class; Accessing Files – FileStream Object, StreamReader class, StreamWriter class, BinaryWriter Class, BinaryReader class;

Unit V: Database and Web Programming ***12Hours + 12 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;

Introduction to Web Services: Creating Simple Web Services

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 = √-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 –
    - Accept deposit from a customer and update his balance.
    - Display the balance
    - Compute and deposit interest
    - Permit withdrawal and update balance
    - Check for the minimum balance, impose penalty, if necessary, and update balance.
34. Create a custom control by enhancing the features of a TextBox control such that when it receives the focus, there is a change in colour from white to some shade of blue and also when it loses the focus, the colour changes to some shade of gray. Use this custom TextBox in a project to take in as input a number and display as output the reversed number.
35. Create a compound control that consists of two TextBoxes and two appropriate labels for the TextBoxes grouped inside a GroupBox. Use this compound control in a project to take in as input two numbers for lower limit and upper limit values and use these values to display all the prime numbers between these limits.
36. Develop an application that will scan all the folders and the sub-folders therein of a particular drive and list them out in a richtextbox. This application should simulate the working of the DOS command DIR /S, the path

of the folders should be displayed in bold. Also make provision for saving the output as a text file in a folder of user's choice and also to print the output.

37. Write a program to store details about books – title, author, no. of pages, in a file. The program should allow the user to search for a book by title and authorname. Also, addition of new books should be allowed.
38. Write a program to implement facilities like creating a files and folders, deleting files and folders, renaming and list all file under a particular folder.
39. Write a program to concatenate two files into a third file.

Unit III

40. 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.
41. 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.
42. 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.
43. 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).
44. Develop a web page that will calculate the monthly installment for a loan amount, given the rate of interest and its term.
45. Develop a web page with a counter that displays the number of visits to the site.
46. 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.
12. 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.
47. 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.
48. Create a web service to add numbers together. Also develop a client program that uses the web service.
49. Create a web service to display all the factors of a positive number. Also develop a client program that uses this web service.
50. Write a web service to connect to a database and retrieve some data from a table and return it to the client.

Instruction to Paper Setter

The question papers will be set according to the following scheme

Theory Questions			Practical Questions			
Unit	**To be set**	**To be answered**	**Unit**	**To be set**	**To be answered**	**Marks**
I	2	1	I	2	1	10
II	2	1				
III	2	1	II	2	1	15
IV	2	1				
V	2	1	III	2	1	15

Recommended Books

Text:

1. **Petroutsos, E.**, *Mastering Visual Basic.NET,* New Delhi: BPB Publications, 2004

Reference:

1. **Peter Aitken's** ," *Visual Basic.NET Programming*" , Deamtech, 2002
2. **Murach's beginning Visual Basic.NET,** *Anne Prince,* Tech Publications PTE Ltd, Singapore, 2002
3. **Steven Holzner** , *Visual Basic .NET Programming, Black Book*, Dreamtech, New Delhi, 2009
4. **Jeffrey R Shapiro**, *Complete Reference, Visual Basic.NET,*
5. **Bill Evjen et al**, *Visual Basic .NET Programming Bible*, IDG Books India (P) Ltd., New Delhi, 2002
6. **Pooja Bembey et al**, *Microsoft Visual Basic .NET Professional Projects,* Prentice Hall of India Pvt. Ltd., New Delhi, 2002
7. **Michael Halvorson**, *Microsoft Visual Basic .NET Step by Step*, Prentice Hall of India Pvt. Ltd., New Delhi, 2002

Paper III : Computer Organization and Architecture

Objective

The objective of this paper is to introduce the student to the concepts of Computer Organization and Architecture. It is assumed that the student has no background in Digital Logic and Boolean algebra.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Marks		
Total	**Total**	**Theory**	**Internal Assessment**	**Total**
120	3	90	10	100

Unit	Topic	Class Hours	Marks
I	Digital Logic	30	18
II	Sequential Logic, Control Unit	30	18
III	Computer Arithmetic	20	18
IV	Input-Output Organization	20	18
V	Memory Organization	20	18
Total		**120**	**90**

Detailed Syllabus

Unit I: Digital Logic 30 Hours

Block Diagram of a digital computer. Bit, Byte, Word. Binary, Decimal, Octal and Hexadecimal Number Systems. Conversion of integers and fractions from one number system to another. 9's, 10's, 1's and 2's Complement. Binary Addition. Subtraction using 9's and 10's complement. Binary Subtraction using 1's and 2's complement. Simplification of Boolean expressions by using Basic Identities of Boolean Algebra viz. X+0=X, X.0=0, X + 1 =1, X.1=X, X+X=X, X.X=X , X+X'=1, X.X'=0, X+Y=Y+X, XY=YX, X+(Y+Z)=(X+Y)+Z, X(YZ)=(XY)Z, X(Y+Z)=XY+XZ, X+YZ=(X+Y)(X+Z) and (X')' = X. De Morgan's Theorem (X+Y)'=X'Y' and (XY)'=X'+Y'.

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. Combinational Circuit. Drawing the logic diagram by using gates to represent a given Boolean expression. Drawing the logic diagram by using gates to represent the simplified Boolean expression. Two Universal gates-NAND and NOR gates. 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.

Three-State Buffer (gate) - Control input and Normal input, High Impedance State, Graphic Symbol.

Canonical Form- Minterms, Maxterms. Standard Form- Sum of Products, Product of Sums.

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

Half-Adder and Full Adder- with Truth Tables, Logic Diagrams and Using Karnaugh maps to derive the Boolean expressions for the Sum and Carry. 4-bit binary adder, 4-bit Incrementer, 4-bit Adder-Subtractor.

3X8 Decoder - with the Truth Table and Logic Diagram. 4X1 Multiplexer - with the Logic Diagram and the Function Table.

Unit II: Sequential Logic, Control Unit **30 Hours**

Flip flop. Present State and Next State of a flip flop. RS, JK, D and T flip flops (Graphic Symbols, Characteristic Tables and Excitation Tables). Indeterminate State of an RS flip flop.

Sequential circuits. Block Diagram of a Sequential circuit. Difference between Combinational and Sequential circuits. Difference between a Flip Flop and a Register, Difference between a Flip Flop and a Counter.

Design of counters using only type of flip flops (viz. JK flip flops only). Design of Counters, Present State and Next State 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.

Two methods to implement a Common Bus System (using Multiplexers, using Three-State Buffers).

Brief definition of the following terms- ALU, Registers, Program Counter (PC), Sequence Counter (SC), Accumulator (AC), Data Register, Address Register, Instruction Register.

Instruction Code, OP Code (Operation Code), Mode field of an Instruction Format, Effective Address. Stored Program Organization.

Instruction Format- Three Basic Computer Instruction Formats viz. Memory-Reference-Instructions (MRI), Register-Reference-Instructions (RRI), Input-Output-Instructions. Instruction Set Completeness. Timing and Control- Clock Pulses, Control Unit and its Block Diagram, Timing Signals, Difference between Hardwired-Control and Microprogrammed-Control.

Phases of an Instruction Cycle- Fetch, Decode, Execute. Flowchart of an Instruction Cycle.

Memory-Reference Instructions- AND- AND to Accumulator, ADD- ADD to Accumulator, LDA- Load to Accumulator , STA- Store from Accumulator to memory, BUN- Branch Unconditionally, BSA- Branch and Save Return Address, ISZ- Increment and Skip if Zero.

Register-Reference Instructions- Clear Sequence Counter, CLA- Clear Accumulator , CLE- Clear E (E is the extended AC bit), CMA- Complement Accumulator (Complement AC), CME- Complement E (E is the extended AC bit), CIR- Circulate the Accumulator to the Right, CIL- Circulate the Accumulator to the Left, INC- Increment Accumulator, SPA- Skip if Accumulator is Positive, SNA- Skip if Accumulator is Negative, SZA- Skip if Accumulator is Zero, SZE- Skip if E is Zero (E is the extended AC bit), HLT- Halt Computer (Clear the Start-Stop flip flop).

Input output configuration, Input Register, Output Register.

Input-Output Instructions- INP -Input character, OUT- Output character, SKI- Skip on input flag, SKO- Skip on output flag, ION- Interrupt Enable On, IOF- Interrupt Enable Off.

Block Diagram of a CPU- (with three major components- viz. Control Unit, ALU, Register Set).

Three-Address Instructions, Two-Address Instructions, One-Address Instructions, Zero-Address Instructions.

Addressing Modes (Implied, Immediate, Register, Register-Indirect, Direct, Indirect, Relative, Indexed).

Unit III: Computer Arithmetic 20 Hours

Sign bit, Signed-Magnitude Representation, Signed-Two's Complement Representation, and Unsigned Numbers.

Arithmetic Left Shift (Multiplication of a Signed-Binary Number by 2, Sign Reversal, Overflow, Boolean expression to detect Arithmetic Shift-Left Overflow, V_s- Overflow Flip Flop). Arithmetic Right Shift (Division of a Signed-Binary Number by 2).

Addition and Subtraction with Signed-Magnitude numbers- Hardware Implementation and Flowchart- Flip Flop E(The Extended Accumulator bit), AVF(Add Overflow Flip Flop), Augend, Addend, Minuend, Subtrahend; Addition and Subtraction with Signed-Two's Complement numbers (Hardware Implementation and Flowchart).

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.

Division Algorithm (Dividend, Divisor, Quotient and Remainder, An Algorithm for Division of Signed-Magnitude Data, Finding both the Quotient and Remainder).

Unit IV: Input-Output Organization 20 Hours

I/O Bus versus Memory Bus, Isolated I/O method versus Memory-Mapped I/O; Strobe Control, Handshaking, Timeout mechanism.

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, Flowchart for CPU Program to input data).

Priority, PI (Priority In), PO (Priority Out). Priority Interrupt -Daisy-Chain Priority Interrupt and Parallel Priority Interrupt. Daisy-Chain Priority Interrupt (One-Stage of the Daisy-Chain Priority Arrangement, VAD -Vector Address, Interrupt Acknowledge). Parallel Priority Interrupt (Priority Interrupt Hardware, Two flip flops –Interrupt Status flip flop IST and Interrupt Enable flip flop IEN, Two Registers - Mask Register and Interrupt Register).

Direct Memory Access (CPU Bus signals for a DMA Transfer, Block diagram of a DMA Controller, DMA Transfer, Word Count Register, Bus Request, Bus Grant, Burst Transfer, Cycle Stealing, Some Applications of DMA).

Unit V: Memory Organization **20 Hours**

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

CAM (Content Addressable Memory or Associative Memory), Associative Memory (Hardware Organization, Block Diagram of Associative Memory, Match Logic, Key Register, Argument Register, Match Register, Masking, Unmasked bits of the argument, Match, No Match); Diagram of an array of Associative Memory of m-words and n-cells per word; Internal Organization of One Cell of Associative Memory; Match Logic for One Word of Associative Memory- List of the Boolean Functions related to the Match Logic. Write and Delete Operations in Associative Memory. Active Word, Inactive Word, and Tag Register.

Cache Memory, Locality of Reference, Hit, Miss, Hit Ratio, Three Mappings in cache viz. Associative Mapping, Direct Mapping (Tag Field, Index Field), Set-Associative Mapping (Set, Two-Way-Set-Associative Mapping). Writing into Cache (Write Through, Write Back), Replacement in Associative Mapping and Set-Associative Mapping, Cache Initialization (Valid Bit).

Instruction for Paper Setter

(The question will 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

Internal Assessment : 10 Marks

Recommended Books

Text:

M. Morris Mano, *Computer System Architecture*, Pearson Education, New Delhi, Third Edition, 1993.

Reference

M. Morris Mano, *Digital Logic and Computer Desig*n, Prentice Hall India, New Delhi, 2004.

Paper IV: 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		
60	2	**External**	**Internal**	**Total**
		40	10	50

Unit	Topic	Minimum Class Hours	Marks
1	Introduction To Software Engineering Software life cycle models	8	8
2	Software project management, requirements and design	18	10
3	Function Oriented Software Design, Object Modelling using UML, Object Oriented Software Development and user interface Design	18	12
4	Testing, software reliability and maintenance	16	10
TOTAL		**60**	**40**

UNIT 1 **8 hours**

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

Software Life cycle models: - Importance of a 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 2 **18 hours**

Software project management: Responsibilities of a software project manager, project planning, project estimation techniques, 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 (Risk identification, risk assessment, Risk Containment), Software configuration Management (Necessity of software Configuration Management, Configuration Management Activities, Source code control system and RCS.

Requirement Analysis and specification: Requirement gathering and analysis, Software requirements specification (Content of the SRS document, Functional Requirements, how to identify the functional requirements, how to document the functional requirements, traceabilty, characteristics of a good SRS document, techniques for representing complex logic).

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

UNIT 3 **14 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), Extending DFD technique to real-time systems, Structured Design (flow chart vs. structure chart, transformation of a DFD model into a structure chart), Detailed Design, Design Review.

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

Object Oriented Software Development: Design Patterns, A Generalized Object-Oriented Analysis and Design Process, OOD Goodness Criteria

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), User interface methodology (Design, a GUI design methodology, Task and object modeling, selecting a metaphor, interaction design and rough layout, user interface inspection).

UNIT 4 **10 hours**

Coding and Testing: coding standards and guidelines, code review (code walkthroughs, code inspection, clean room testing, software documentation), 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, debugging, program analysis tools, integration testing, system testing.

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

Software Maintenance: Characteristics of software maintenance (types of software maintenance, characteristics of software evolution, special problems associated with software

maintenance), software reverse engineering, software maintenance process models, estimation of maintenance cost.

Software Reuse: What can be Reused?, Why Almost no Reuse so far?, Basic Issues in Any Reuse Program, A Reuse Approach

Instruction to Paper Setter

Unit	Questions	
	To be Set	To be Answered
I	2	1
II	2	1
III	2	1
IV	2	1

Recommended Books

Text:

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

References:

2. **Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli**, ***Fundamentals Of Software Engineering***, Second Edition, Prentice Hall of India Private Limited, New Delhi, 2002.
3. **Richard E Fairley**, ***Software Engineering Concepts***, Tata McGraw Hill Publishing Company Limited, New Delhi, 1997.

Paper V: 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	
Honors	Pass	Honors	Pass	Honors	Pass	Honors	Pass
180	100	50	40	20	10	70	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

. 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

	Honours	General
Analysis	8	6
Design	7	4
Implementation	15	10
Internal	20	10
Project Report	10	10
Viva	10	10
Total	**70**	**50**

Paper VI: 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. The students are also provided with an overview of some of the emerging database technologies and applications.
The tutorial classes are to be utilized for discussing case studies.

Outline of the Course

Minimum Class Hours			Exam Time (Hours)	Marks		
Class	**Tutorial**	**Total**		**Theory**	**Internal**	**Total**
95	25	120	3	90	10	100

Unit	Topic	Minimum Class Hours			Marks
		Theory	**Tutorial**	**Total**	
I	Introduction and Conceptual Data Modeling	20	5	25	20
II	Relational Data Model and SQL	20	5	25	20
III	Functional Dependencies and Normalization	10	5	15	10
IV	File Organization	15	5	20	15
V	Query Processing, Transaction Processing,Concurrency Control and Security	20	5	25	20
VI	Advanced Database Concepts and Emerging Applications	10	0	10	5
Total		**95**	**25**	**120**	**90**

Detailed Syllabus

Unit I: Introduction **and Conceptual Data modeling** **25 hours**

Introduction: Introduction to databases, characteristics of the database approach, database users and designers, role of a DBA, advantages of using a DBMS, data models, schemas, instances, DBMS architecture (Three-Schema Architecture), Database systems- Network, Hierarchical, Relational, Data Independence

Conceptual Data Modeling: Phases of database design, 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 (only concepts)

Unit II: **Relational Data Model and Structured Query Language** **25 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, Case study

Relational Algebra: basic relational algebra operations-SELECT, PROJECT, UNION, INTERSECTION, SET DIFFERENCE, Cartesian PRODUCT, JOIN, Aggregate functions

SQL: Characteristics of SQL, Data types in SQL, Types of SQL commands

Data Definition Commands: CREATE SCHEMA, CREATE TABLE, DROP TABLE, ALTER TABLE

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

Data changing commands: INSERT, DELETE, UPDATE

Unit III: **Functional Dependencies and Normalization** **15 hours**

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

Unit IV: **File Organization** **20 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), Operations on Files (Open, Reset, Find, Read, Delete, Modify, Insert, Close), primary methods of file organization -Heap Files, Sorted Files, Hashed Files.Types of Single-level Ordered Indexes (Primary Indexes, Clustering Indexes, Secondary Indexes), Multilevel Indexes: Basic technique, Multilevel indexing using B tree and B+ tree,Indexing on multiple keys

Unit V: **Query Processing, Transaction Processing, Concurrency Control and Security** **25 hours**

Query Processing: Overview of query processing, translation of SQL queries into relational algebra

Transaction Processing: Transaction, ACID properties of transaction, transaction states, schedules, serializability, tests for serializability, recoverability, transaction definition in SQL.

Concurrency Control: Concurrent execution of transaction, Lock-based techniques for concurrency control-Two-Phase locking protocol and its variations, Graph-based protocol, Timestamp based protocol, Deadlock, Deadlock prevention methods, Deadlock detection Deadlock recovery

Security: Risks to data security, role of the DBA in maintaining database security, access protection, encryption, database audits

Unit VI: Advanced Database Concepts and Emerging Applications **10 hours**

Introduction to Object-Oriented Databases, Distributed databases, Client-Server Architecture, Data Mining, Data Warehousing, Deductive databases, Databases on the World Wide Web, Multimedia Databases, Geographical Information Systems

Sample Case Studies

Case Study I: DreamHouse Properties

The requirement collection and analysis phase of the database development lifecycle was carried out at a DearmHouse branch. The requirements specification, which describes the information to be held in the DeramHouse database and the transactions required by the supervisor are as follows.

Data Requirements

1. Each branch of DreamHouse has staff who are dedicated to the management of property for rent. The staff works in groups that are supervised by a Supervisor and supported by a Secretary.
2. The information stored on each branch office includes a unique branch number, address (street, area, city, pincode), telephone number and fax number.
3. The information held on all members of staff includes a staff number, name (first and last), address, telephone, sex, date of birth, job title and the number and address of the branch office at which they work. Additional information held on staff with the job title of Secretary is their typing speed.the staff number is unique across all branches of the DreamHouse company.
4. Each Supervisor supervises the day-to-day work of a group of staff(minimum 5 to a maximum of 10 members of staff, at any one time)
5. A portfolio of property for rent is available at each DreamHouse branch. Each property for rent is managed by a particular member of staff. A member of staff may manage a maximum of 10 properties for rent at any one time.The information stored on each property for rent includes : the property number, address(street,area,city,pincode),type,number of rooms,monthly rent, and the name and address of the property owner.The monthly rent for a property is reviewed annually.Most of the property rented out by DreamHouse are flats.Each property is owned by a single owner,
6. The details of owners of property are also stored. There are two main types of property owner: private owner and business owner. The information stored on private owners includes the owner number, name (first, last), address and telephone number. The information stored on business owners includes the owner number, name of business, business type, address, and telephone number and contact name. Each owner owns at least one property.
7. The staff responsible for the management of property for rent must undertake the following activities :
   a) To ensure that property is rented out continuously.This may require placing an advertisment describing a property for rent in an appropriate newspaper. The information stored on each advert includes the advert number, the date the advert was placed in the newspaper, the name of the newspaper, the cost, and some details of the property including the property number, type and address. The advert number is unique across all DeramHouse branches. The information stored on each newspaper includes the newspaper name, address, telephone no., fax no. and contact name. Properties are only advertised in the newspapers if they prove difficult to rent.
   b) To set up interviews with clients interested in renting property. The information stored as a result of each interview includes the date of the interview and any general comments about the client. During the interview, the details of the clients are also collected.

However, some clients do not attend an interview and simply provide their details by telephone or on their first visit to a DreamHouse branch. The information stored on clients includes the client number, name (first and last), current address, telephone number and some information on the desired property, including the preferred type of accommodation, and the maximum rent the client is prepared to pay. The client number is unique across all DreamHouse branches.

c) To encourage clients to view properties for rent. The information stored includes the client's number, name and telephone no., the property no and address, the date the client viewed the property and any comments made by the client regarding the suitability, or otherwise, of the property. A client may view the same property only once on a given date.

d) To organize the lease agreement between a client and a property. Once a client agrees to rent a property, a lease agreement is organized by a member of the staff. The information on the lease agreement includes the lease no., the client no. and name, the property no, address, type and no. of rooms, the monthly rent, and method of payment, deposit (calculated as twice the monthly rent), whether the deposit is paid, and the date the rent period starts and finishes,the duration of the lease. The lease no. Is unique across all DreamHouse branches. A client may hold a lease agreement associated with a given property for a minimum of three months to a maximum of 1 year.

e) To carry out inspections of property on a regular basis to ensure that the property is correctly maintained. Each property is inspected at least once over a six month period. However, DreamHouse staff only carry out inspections of property that is currently being rented or is available for rent. The information stored on the inspection includes the property number and address, date of the inspection, name of the member staff who carried out the inspection, and any comments on the condition of the property.

Transaction Requirements

The main transactions require by Supervisors include:

(a) Produce a list of staff supervised by a Supervisor.
(b) Produce a list of staff supported by a Secretary.
(c) Produce a list of Supervisors at each branch.
(d) Create and maintain records recording the details of property for rent and the owners at each branch.
(e) Produce a report listing the details of property (including the rental deposit) at each branch.
(f) Produce a list of properties managed by a specific member of staff.
(g) Create and maintain records describing the details of clients at each branch.
(h) Produce a list of clients registered at each branch.
(i) Search for properties that satisfy various criteria.
(j) Create and maintain records holding the details of viewings of properties made by clients.
(k) Produce a report listing the comments of clients concerning a specific property.
(l) Create and maintain records detailing the adverts for a specific property.
(m) Produce a list of all adverts for a specificproperty.
(n) Produce a list of all adverts place in a specific newspaper.
(o) Create and maintain records describing the details of lease agreements between a client and a property.

(p) List the details of the lease agreement for a specific property.
(q) Create and maintain records describing the details of inspections of properties.
(r) Produce a list of all inspections of a specific property.

Case Study II: Georgia Tech Library

The Georgia tech Library (GTL) has approximately 16,000 members, 100,000 titles and 250,000 volumes. About 10 percent of the volumes are out on loan at any one time. The librarians ensure that the books that members want to borrow are available when the members want to borrow them. Also, the librarians must know how many copies of each book are in the library or out on loan at any given time. A catalog of books is available on-line that lists books by author, title and subject area. For each title in the library, a book description is kept in the catalog that ranges from one sentence to several pages. The reference librarians want to be able to access this description when members request information about a book. Library staff is divided into chief librarian, departmental associate librarians, reference librarians, check-out staff, and library assistants. Books can be checked out for 21 days. Members are allowed to have only five books at a time. Members usually return books within three to four weeks. Most members know that they have one week of grace before a notice is sent to them, so they try to get the book returned before the grace period ends. About 5 percent of the members have to be sent reminders to return a book. Most overdue books are returned within a month of the due date. Approximately 5 percent of the overdue books are either kept or never returned. The most active members of the library are defined as those who borrow at least ten times during the year. The top 1 percent of membership does 15 percent of the borrowing and the top 10 percent of membership does 40 percent of the borrowing. About 20 percent of the members are totally inactive in that they are members but do never borrow. To become a member of the library, applicants fill out a form including their SSN, campus and home mailing addresses and phone numbers. The librarians then issue a numbered, machine readable card with the member's photo on it. This card is good for four years. A month before the card expires, a notice is sent to a member for renewal. Professors at the institute are considered automatic members. When a new faculty member joins the institute, his or her information is pulled from the employees' record and a library card is mailed to his or her campus address. Professors are allowed to check out books for three-month intervals and have a two-week grace period. Renewal notices to professors are sent to the campus address. The library does not lend some books, such as reference books, rare books and maps. The librarians must differentiate between books that can be lent and those that cannot be lent. In addition, the librarian have a list of some books they are interested in acquiring but cannot obtain, such as rare or out-of-print books and books that were lost or destroyed but have not been replaced. The librarian must have a system that keeps track of books that cannot be lent as well as books they are interested in acquiring. Some books may have the same title; therefore, the title cannot be used as a means of identification. Every book is identified by its International Standard Book Number (ISBN), a unique international code assigned to all books. Two books having the same title can have different ISBNs if they are in different languages or have different bindings

(hard cover or soft cover). Editions of the same book have different ISBNs. The proposed system must be designed to keep track of the members, the books, the catalog, and the borrowing activity.

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
VI	2	1

Recommended Books

Text:

1. **R.Elmasri, S.B Navathe**, *Fundamentals of Database Systems*, Addision, Wesley (Third Edition) 2000.

Reference:

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.

Paper VII: Data Communication and Networks

Objectives

Data communication and networks have become an integral part of our lives. This paper intends to give an in depth knowledge about the various layers in the networks, bringing to the fore the various issues involved in the design of such layers, and the various algorithms used to resolve them. There is also an insight into the vistas of data communication. It also introduces the students in the newer areas of computer networking.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Marks		
Theory	Theory	Theory	Internal	Total
120	3	90	10	100

Unit	Topic	Minimum Class Hours	Marks
I	Introduction to Computer Networks, Physical Layer	20	15
II	Data Link Layer	20	15
III	Medium Access Control Sublayer	20	15
IV	Network Layer	20	15
V	Transport Layer	20	15
VI	Application Layer, WWW, Network Security	20	15
Total		**120**	**90**

Detailed Syllabus

Unit 1: Introduction to computer Networks and Physical Layer 20hrs

Introduction to computer Networks : Use of computer Networks; Wired and wireless networks; Types of networks – LAN, MAN, WAN; Network topology; OSI Reference Model- Outline, Protocol Hierarchies, Design considerations; TCP – IP Reference Model; Comparison between the two reference models; ATM Reference Model; Examples – Internet, X.25, Frame Relay, Ethernet, Wireless LANS.

Physical Layer: Fourier Analysis (Qualitative), Maximum data rate of a channel, Bit rate and Baud rate; Baseband and Broadband; Guided Transmission Media – Magnetic, Twisted pair, Coaxial cable, Fibre Optics; Wireless transmission, Electromagnetic Spectrum, Radio transmission, Microwave Transmission, Infrared transmission, Communication Satellites- Geo,

Meo, and Leo Satellites, Satellites versus Fibre Frequency Division and Time Division Multiplexing; circuit, Message and Packet Switching; outline of PSTN, Mobile telephone system – First, Second and Third Generation mobile phones; Cable Television – Internet over Cable.

Unit 2: Data Link Layer **20 hrs**

Design Issues – Services provided to the higher layer, Framing, Error control, Flow Control; Error Detection and Correction – error correcting Codes, error detecting codes; Elementary Data Link Protocols – Unrestricted simplex protocol, simplex stop-and – wait protocol, protocol for Noisy Channel; Sliding Window protocols – one bit sliding window, Go Back n protocol, protocol using Selective Repeat, Examples – HDLC, Data Link Layer in the Internet, PPP

Unit 3 Medium Access Control Sublayer **20 hrs**

Channel Allocation Problem – Static and Dynamic channel allocation; Multiple access- Aloha, CSMA; Collision free protocols; Wireless LAN protocols; IEEE standard 802.3 (Ethernet) – Cabling, Encoding, MAC Sublayer, Switched Ethernet, Fast Ethernet, Gigabit Ethernet; IEEE standard 802.11 (Wireless LANS) – Protocol stack, physical Layer, MAC sublayer, Frame Structure; IEEE standard 802.16 (Broadband Wireless) – protocol stack, Physical Layer, MAC Sublayer, Frame Structure; Bluetooth –Architecture, Application, Protocol Stack, Radio Layer, Baseband Layer, Frame structure; Bridges – spanning tree bridges, Remote bridges.

Unit 4: Network Layer **20hrs**

Design Issues- Store and forward packet switching, Services provided to higher layer, Connection Oriented and Connectionless services, Virtual circuits and Datagram subnets; Routing algorithms- Shortest Path Routing, Flooding, Distance Vector Routing, Link State Routing, Hierarchical Routing, Broadcast Routing, Multicast Routing, Routing for Mobile hosts, Routing for Adhoc networks; Congestion Control Algorithms – General principles, Load shedding, Jitter control, QoS- Requirements, Techniques to achieve QoS, RSVP; Internetworking- How networks can be connected, Tunnelling, Fragmentation; Internet Protocol – IP addresses, Subnets, CIDR, Network Address Translation; Internet Control Protocol – ARP, RARP, BOOTP, DHCP; Mobile IP – Routing.

Unit 5: Transport Layer **20 hrs**

Design Issues, Services presented to higher layers; Transport protocols – Addressing, Connection Establishment and Release, Flow Control and Buffering, Multiplexing, Crash Recovery, Simple Transport Protocol

Internet Transport Protocols: UDP – Remote Procedure Call, Real-time transport Protocol; TCP – Service Model, Protocol, Header, Connection Establishment and Release, Connection Management, Transmission Policy, Congestion Control; Wireless TCP and UDP; Performance Issues

Unit 6: Application Layer, World Wide Web and Network Security 20hrs

Application Layer: Domain Name System – Name space, Resource Records, Name Servers, Electronic Mail – Architecture and Services, user agent, Message formats – MIME, Message Transfer – SMTP, Message Delivery - POP3 and IMAP, Webmail, Telnet, FTP

World Wide Web: Architectural Overview, Client Side, Server Side, Uniform Resource Locators, Statelessness and Cookies; Hypertext Transfer Protocol; Wirelesses Web – Wireless Application Protocol, I-Mode

***Network Security*:** Cryptography, Substitution Ciphers, Transposition Ciphers, One time pads, Quantum Cryptography, Cryptographic principles; Symmetric Key Algorithms – Data Encryption Standard, Advanced Encryption Standard; Public Key Algorithms – RSA; Digital Signatures – Symmetric Key, Public Key, Message digest, Birthday Attack; Communication Security – firewalls, Virtual Private Networks, Wireless Security – 802.11 security, Bluetooth Security, WAP Security; Authentication Protocols – Based on shared secret key, Deffie-Hellman Key Exchange, Key Distribution Centre, Kerberos, Public Key

Instruction to Paper Setter

(The question papers will 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
VI	2	1

Recommended Books

Text

1. **Andrew S. Tenenbaum**, *Computer Networks* (Fourth Ed.), Prentice Hall of India, 2002

Reference

1. **William Stallings**, *Data and Computer Communications* (Sixth Ed.), Prentice Hall of India, 2000
2. **U. D. Black**, *Data Communications and Distributed Networks* (3rd Ed.), Prentice Hall of India, 1993
3. **Fred Halsall**, *Data Communication, Computer Networks and Opens Systems*, (4th Ed.), Pearson Education, 2000
4. **William Stallings**, *Cryptography and Networking Security - Principles and Practice, Pearson Education, 2000*

Paper VIII
Elective: I Web Programming in PHP and MySQL

Objective: The course is designed to impart knowledge and skill to the students to learn and know the principles and techniques of programming for the Web. This course aims to build in the students a web programming knowledge using PHP and MySQL.

Outline of the Course:

Minimum Class Hours			Exam Time (Hours)		Marks			
Theory	Practical	Total	Theory	Practical	Theory	Practical	Internal	Total
60	60	120	2	3	50	40	10	100

Detailed Syllabus:

Unit	Topic	Minimum Class Hours			Marks Theory
		Theory	Practical	Total	
I	**Database Applications and the Web, Introduction to PHP, Arrays, Strings and Advanced Data Manipulation in PHP**	12	15	27	10
II	**Introduction to Object Oriented Programming with PHP5, Advance Features of Object Oriented Programming**	18	15	33	15
III	**Sql and MySQL, Querying Web Database, Writing to Web Database, Reporting, Validation with PHP and Javascript**	18	15	33	15
IV	**Sessions, Authentication and Security, Errors, Debugging and Deployment**	12	15	27	10

Unit-I

Introduction to HTML, HTTP and other Web Protocols. Database Applications and the Web: The Web, Three Tier Architectures.
Introduction to PHP: Introducing PHP, Conditions and Branches, Loops, Functions, Working with types, User-defined Functions, A Working Example
Arrays, Strings and Advanced Data Manipulation in PHP: Arrays, Strings, Regular Expressions, Dates and Times, Integers and Floats

Unit-II

Introduction to Object-Oriented Programming with PHP 5: Classes and Objects, Inheritance, Throwing and Catching Exceptions
Advance Features of Object-Oriented Programming in PHP 5: Working with Class Hierarchies, Class Type Hints, Abstract Classes and Interfaces

Unit-III

SQL and MySQL: Database Basics, MySQL Command Interpreter, Managing Databases and Tables, Inserting, Updating and Deleting Data, Querying with SQL SELECT, Join Queries
Querying Web Databases: Querying a MySQL Database using PHP, Processing User Input, MySQL Function Reference
Writing to Web Databases: Database Inserts, Updates and Deletes, Issues in Writing Data to Database
Reporting: Creating a Report, Producing PDF, PDF-PHP Reference
Validation with PHP and Javascript: Validation and Error Reporting Principles, Server-Side Validation with PHP, Javascript and Client-Side Validation

Unit-IV

Sessions: Introducing Session Management, PHP Session Management, Case Study: Using Sessions in Validation, When to Use Sessions, PHP Session API and Configuration
Authentication and Security: HTTP Authentication, HTTP Authentication with PHP, Form-Based Authentication, Protecting Data on the Web
Errors, Debugging and Deployment: Errors, Common Programming Errors, Custom Error Handlers

Theory Questions		
Unit	**To be set**	**To be Answered**
I	2	1
II	2	1
III	2	1
IV	2	1

Recommended Book:

Hugh E Williams and David Lane, ***Web Database Applications with PHP and MySQL,*** O'Rielly 2nd Edition

References:

Luke Welling and Laura Thomson, ***PHP and MySQL Web development,*** Sams Publishing 2nd Edition

Paper VIII
Elective-2: Object Oriented Programming through Java

Objective

The course is designed to impart knowledge and skill required to solve the real world problem using object-oriented approach utilizing Java language constructs. This course covers the two main part 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 Datatypes.
- 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.
- Setting up GUI using AWT/Swing.
- 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	Total	Theory	Practical	Theory	Practical	Internal	Total
60	**60**	**120**	**2**	**3**	**50**	**40**	**10**	**100**

Unit	**Topic**	**Minimum Class Hours**			**Marks**
		Theory	**Practical**	**Total**	**Theory**
I	Introduction to Java Programming, Classes and Methods	12	12	24	10
II	Inheritance, Exception handling, Multithreading Enumerations and Autoboxing	12	12	24	10
III	Generics, String handling, java.lang and Input/Output	12	12	24	10
IV	Images, Applet class and Swing	12	12	24	10
V	Networking, JDBC, Java Servlets	12	12	24	10
Total		60	60	120	50

Detailed Syllabus

Unit I: Introduction to Java Programming, Classes and Methods 24 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 24 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( ), suspend( ) and resume( ), Deadlock

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 and Input/Output 24 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, Raw types and legacy code, Ambiguity errors, Generic Restrictions

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, Serialization (Serializable, Externalizable), ObjectOutputStream, ObjectInputStream

Unit IV: Images, Applet class and Swing 24 Hours

Images: File formats, Image fundamentals, Creating, Loading and Displaying
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
Swing: Components and Containers, Event Handling, Creating a Swing applet, text field, Swing buttons(JButton, check boxes and radio buttons), scrolling, list, combo boxes, tabbed panes, layouts (FlowLayout, BorderLayout, GridLayout), tables and trees

Unit V: Networking, JDBC, Java Servlets 24 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

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 factorial of positive integer using recursion.
6. Write a Java program to accept the 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 content 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 an applet using swing to display four push buttons and a text field. Each button displays an icon that represents the flag of a country. When a button is pressed, the name of the country is displayed in the text field. The applet begins by getting its context page and setting the layout manager of that pane. Next, the applet is registered to receive action events that are generated by the buttons. A text filed is then created and added to the applet. Finally, a handle for action event displays the command string that is associated with the button. The text field is used to present this string.
23. Create an applet using swing that displays four check boxes and a text field. When a check box pressed. Its text is displayed in the text field. The context pane for the JApplet object is obtained, and a flow layout is assigned as its layout manager. Next, four check boxes are added to the context pane, and icons are assigned for the normal, roll over and selected status. The applet is then registered to receive item events. Finally, a text field is added to the context pane.
24. Create an applet using swing that displays four radio buttons and one text field. When a radio button is selected, its text should be displayed in the text field.
25. Create an applet using swing to display a tabbed pane. The first tab is titled "Cities" and contains four buttons. Each button displays the name of a city. The second tab is titled "Color" and contains three check boxes. Each check box displays the name of a color. The third tab is titled "Flavors" and contains one combo box. This enables the user to select one of three flavors.
26. Create an applet using swing to demonstrate a scroll pane. First, the context pane of the JApplet object is obtained and a border layout is assigned as its layout manager. Next a JPaned object is create and four hundred buttons are added to it, arranged into twenty columns. The panel is then added to a scroll pane, and the scroll pane is added to the context pane. This causes vertical and horizontal scroll bars to appear. You can use the scroll bars to scroll the button into view.
27. Designed an applet using swing to create a tree and recognize mouse clicks on it. The init() method gets the context pane for the applet. A DefaultMutableTreeNode object labeled "Options" is created. This is the top node of the tree hierarchy. Additional tree nodes are then created, and add() method is called to connect these nodes to the tree. A reference to the top node in the tree is provided as the argument to the JTree constructor. The tree is then

provided as the argument to the JScrollPane constructor. The scroll pane is then added to the applet. Information absent mouse click events is presented in this text field.

28. Create an applet using swing to create and use a table. The context pane of the JApplet object is obtained and a border layout is assigned as its layout manager. A one-dimensional array of strings is created for the column headings. This table has three columns. A two-dimensional array of strings is created for the table cells. You can see that each element in the array is an array of three strings. These arrays are passed to the Jtable constructor. The table is added to ascroll pane and then the scroll pane is added to the context pane.
29. 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.
30. 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".
31. 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].
32. Write a Java program that prints the addresses and names of the local machine and two well known explored Internet web sites.
33. Implement a simple networked communications clients and server. Message are typed into the window at the server and written across the network to the client side, then they are displayed to demonstrate datagrams.

Instructions to Paper Setter

Questions should be set according to the following scheme.

Theory Questions			**Practical Questions**		
Unit	**To be set**	**To be Answered**	**To be set**	**To be Answered**	**Marks**
I	2	1	2	1	14
II	2	1			
III	2	1	3	2	26
IV	2	1			
V	2	1			
Total	**10**	**5**	**5**	**3**	**40**

Recommended Books

1. **Herbert Schildt**, *The Complete Reference Java* , Tata McGraw Hill Seventh Edition 2007.

References:

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

Paper IX

Operating Systems and Introduction to Linux

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. The students will also learn about the Linux operating system, which is a full-blown UNIX clone and is fast gaining popularity worldwide.

N.B. Adequate Linux practical classes should be conducted and the same be evaluated as part of internal assessment. However there will be no external examination for Linux practical.

Outline of the Course

Minimum Class Hours	Exam Time (Hours)	Marks		
		External	Internal	Total
120	3	70	10	80

Unit	Topic	Minimum Class Hours	Marks
I	Concepts & Processes	18	10
II	Memory Management	24	15
III	File Systems	18	10
IV	Input/Output	18	10
V	Deadlocks	18	10
V.	Introduction To Linux	24	15
Total		120	70

Unit I: Concepts & Processes **18 hours**

Operating system objectives and functions, Operating system as a Resource manager, Operating system concepts (Files, Deadlocks, Memory Management, Input/Output, Processes, The Shell, Security, System Calls for Process Management), The evolution of Operating Systems (Serial Processing, Simple Batch Systems, Multiprogrammed Batch Systems, Time Sharing Systems,

Real Time Systems), Introduction to Processes (The Process Model, Process Creation, Process Termination, Process Hierarchies, Process States, Implementation of Processes, Process Control Block), Interprocess Communication (Race conditions, Critical Sections, Mutual Exclusion with Busy Waiting, Sleep and wakeup, Semaphores, Event Counters, Monitors, Message Passing), Classical IPC problems (The Dining Philosophers Problem, The Sleeping Barber Problem), Process Scheduling (Round Robin Scheduling, Priority Scheduling, Multiple Queues, Shortest Job First, Two level Scheduling).

Unit II: Memory Management **24 hours**

Memory management without swapping or paging (Monoprogramming without swapping or paging, Multiprogramming and memory usage, Multiprogramming with fixed partitions), Swapping (Multiprogramming with variable partitions, Memory management with linked lists, Memory management with the Buddy system), 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), Modeling Paging Algorithms (Belady's Anomaly, Stack Algorithms, The Distance string, Predicting page fault rates), Design issues for Paging Systems (Page size, shared pages), Implementation issues(Operating System involvement with Paging, Page Fault Handling, Locking pages in Memory), Segmentation (Implementation of pure segmentation, Segmentation with Paging: MULTICS).

Unit III: File Systems **18 hours**

Files (File Naming, File structure, File types, File access, File attributes, File operations, Memory mapped files), Directories (Hierarchical directory systems, Path names, Directory operations, Implementing directories, Shared files, Disk space management, File system reliability, File system performance), Security (The security environment, Generic Security Attacks, Design Principles For Security, User Authentication), Protection mechanisms (Protection Domains, Access Control Lists, Capabilities, Protection Models, Covert Channels), Type of File Systems (FAT, VFAT, FAT32, NTFS).

Unit IV: Input/Output **18 hours**

Principles of I/O hardware (I/O devices, Device Controllers, Direct memory access), Principles of I/O software (Goal of the I/O software, 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, Track-at-a-time caching, RAM disks) Clocks (Clock hardware, Clock software), X-Window system.

Unit V: Deadlocks **18 hours**

Resources, Deadlock (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 trajectorics, 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), other issues (two-phase locking, non-resource deadlock, starvation).

Unit VI: Introduction to Linux **24 hours**

General overview of the system (Unix and its clones, System Architecture), Features of Linux (Simplicity, Reusable components, Filters, Flexibility, Open source code and freeware, Multitasking and Multiuser capability, Security, Internet tools, X Window GUI), Shell (Introduction, Bourne Shell, C Shell, Korn Shell, GNU Bourne Again Shell), Kernel (Introduction, Block diagram of system kernel), Introduction to vim editor (start vim, the three modes, create, save and open a text file, positioning by character, positioning by line, positioning by word, positioning in the word, positioning on a numbered line, inserting text, deleting text), Linux commands (date, cal, ls, mkdir, rmdir, rm, pwd, cp, cat, mv, shell, man, who, sort, at, echo, grep, find, tail, head, chmod, I/O redirection and piping), Shell scripts (read, exit status of a command, positional parameters, exit, if, case, expr, while, until, for, break, continue, select), Miscellaneous (telnet, ftp, GNOME).

Suggested Practical Assignments

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. Write a script that will accept a filename from the keyboard and determine whether the file exists. If the file exists then its contents will be displayed else an error message will be displayed.

4. 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.

5. Write a shell script that displays a list of all files in the current directory to which you have read, write and execute permissions.

6. 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.

7. 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*.

8. Write a script that will accept a string from the keyboard and echo a suitable message if it doesn't have at least 10 characters

9. 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.

10. Write shell script to convert file names from UPPERCASE to lowercase file names or vice versa.

11. 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.

12. Create a script that waits for somebody to login, then prints an appropriate message.

13. 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.

14. Write a shell script to print contents of file from given line number to next given number of lines. For e.g. If we called this script as Q14 and run as $ Q14 5 5 *filex* , Here print contains of '*filex*' file from line number 5 to next 5 line of that file.

15. Write a shell script to display the current date and determine if it is a leap year or not.

16. Write a shell script to display a menu on the screen to execute any one of the commands as given below:
    a. clear the screen
    b. show list of files in current working directory
    c. display contents of a given file, where filename is entered by the user
    d. Merge the contents of two given files into a new file

Instruction for Paper Setter

The question paper will 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
VI	2	1

Recommended Books

Text

1. **Andrew S Tanenbaum**, *Modern Operating Systems* , (Second Ed.), Pearson Education, New Delhi
2. **Sumitabha Das**, *UNIX Concepts & Applications*, (Second Ed.), Tata McGraw Hill, 2001

Reference

1. **William Stallings**, *Operating Systems*, (Fourth Ed.), Prentice Hall of India, 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.
4. **Bill Ball and David Pitt**, *Red Hat Linux 7 unleashed*, Techmedia Publication
5. **Maurice J. Bach**, *The Design of the Unix Operating System*, Prentice Hall of India, New Delhi.
6. **Kernighan and Pike**, *The Unix Programming Environment*, Prentice Hall of India, New Delhi.