1. **Number Systems and Codes**
Introduction to positional number system, signed magnitude numbers, floating point numbers, binary arithmetic: addition, subtraction, multiplication and division, Base conversion, conversion formulas with examples, one’s and two’s compliment arithmetic,

Computer codes – BCD codes, gray codes, excess-3 codes, parity checks, Hamming and alphanumeric codes.

2. **Digital Logic Families**
Qualitative introduction to digital ICs, TTL, Schottky TTL, ECL, MOS Logic, CMOS Logic, Tri-state logic: Characteristics and properties.

3. **Combinational Logic Design**
Introduction, standard representations for logical functions, Karnaugh map representation, simplification of logical functions using K-map, minimization of logical functions specified in minterms/maxterms or Truth Table, minimization of logical functions not specified in minterms/maxterms, Don’t care conditions, design examples, Ex-or and Ex-nor simplification of K-maps, five and six-variable K-maps, QM method, MEV method.

**Combinational Logic Design using MSI circuits**
Introduction, multiplexers and their use in combinational logic design, demultiplexers/decoders and their use in combinational logic design, adders and their use as subtractors, digital comparators, parity generators/checkers, code converters, priority encoders, 7-segment decoder/driver.

**Synchronous Sequential Circuits**
Introduction, FSM model, memory elements and their excitation functions. Synthesis of synchronous sequential circuits, capabilities and limitation of FSM, state equivalence and minimization, simplification of incompletely specified machines.

**Asynchronous Sequential Circuits**
Fundamental mode circuits synthesis, state assignment, pulse mode circuits.

**A to D and D to A Converters**
Introduction, Study of different types of analog to digital and digital to analog converters, their resolution, conversion time, sensitivity accuracy and other parameters. Study of some commercially available ADC and DAC chips.

**BOOKS**
1. R.P. Jain: Modern Digital Electronics, TMH.
2. Z. Kohavi: Switching and Finite Automata Theory, TMH
3. M.M. Mano: Digital Logic Design, PHI.
Object Oriented Programming using C++

IT-203

Theory: 60
Sessional: 40

1. **Object Oriented Programming and Design**


2. **C++ Programming Basics**

Fundamentals: Variables and assignments, Input and Output, Data Types and Expressions, Flow of control, Subprograms: Top down design, Predefined functions, Programmer defined functions, Procedural abstractions, Local variables, Overloading function names, Operator overloading, Parameter passing, this pointer, Destructors, Copy constructors, Overloading the assignment operator, Virtual functions, Function Calling functions, Friend functions, Recursive function, Recursive member function.

3. **C++ Object Oriented Concepts**

Objects and Classes: Use of file for I/O, Formatting output with stream functions, Character I/O, Inheritance, Structures for diverse data, Structures as function arguments, Initializing structures, Defining classes and member functions, Public and private members, Constructors for initializations, Standard C++ classes, Derived classes, Flow of Control, Use of Boolean expressions, Multiway branches, Use and design of loops.

4. **C++ Data Structures and Advanced Topics**

Arrays – Programming with arrays, arrays of classes, arrays as function arguments, Strings, Multidimensional arrays, Arrays of strings, Pointers, Dynamic arrays, Classes and dynamic arrays, Base classes, access control, Templates – generic classes and functions, namespaces.

**BOOKS**
1. Herb Schildt: C++ - The Complete Reference, TMH, Delhi
3. Mastering C++, K.R. Venugopal, TMH, New Delhi
Data Structures

IT-205

L     T  Theory:  60
4     1  Sessional: 40

Note: All implementations in C language.

1. **Introduction:**
Introduction: Internal representation, integers, floating point numbers, packed decimal, characters, data types and data object, fundamentals of pointers in C, pointer declaration, passing pointer to functions, pointers and 1-d arrays, dynamic memory allocation, operation on pointers, pointers and 2-d arrays; Files and related operations in C.

2. **Searching and Sorting Techniques**
Efficiency of algorithms in terms of time and storage requirements, O-notation, Searching techniques: Linear and Binary, Sorting techniques: Selection, Bubble, Insertion, Mergesort, Quicksort and Radix sort

3. **Simple Data Structures**
Arrays: axiomatic definition of array, representation of array in storage, address mapping function, access table method of storage of arrays, sparse arrays, manipulation transpose, addition multiplication of sparse matrices, examples for application of stacks, expression evaluation, mazing problem, sequential allocation for stacks and queues; multiple stacks and queues.

4. **Linked Data Structures**
Linked Lists; definition, allocation for stacks and queues. Examples of linked lists, polynomial addition, comparison of sequential and linked allocation of storage; inversion, concatenation & copying of the lists.

Doubly Linked List: Definition of circular and doubly linked list, header node, insertion and deletion, sparse matrix, representation using doubly linked lists. Examples for application of doubly linked lists; dynamic storage management; node structures, routines for allocation and deallocation, generalized lists and recursive algorithms for copying and comparison of lists.

5. **Advanced Data Structures**
Trees, Basic concepts and definitions of a tree and binary tree and associated terminology, examples of tree structures. Binary trees traversal, Binary tree representation of trees, transformation of trees into binary trees, some more operations on binary trees. Graphs: Representation of graphs and their traverseral.
BOOKS
3. R.L. Kruse: Data Structures & Program Design in C, PHI.
Web Site Design

IT-207

L T Theory: Sessional: 60 40
3 1

1. Information Architecture


2. Dynamic HTML and Web Designing


3. CGI using PERL


4. Java Server Pages

Basics, Integrating Scripts in JSPs, JSP Objects and Components, configuring and troubleshooting, JSP: Request and response objects, Retrieving the contents of a an HTML form, Retrieving a Query String, Working with Beans, Cookies, Creating and Reading Cookies. Using Application Objects and Events.

5. XML

Relationship between HTML, SGML and XML, Basic XML, Valid Documents, Ways to use XML, XML for Data Files, Embedding XML into HTML documents, Converting XML to HTML for DISPLAY, Displaying XML using CSS and XSL, Rewriting HTML as XML, The future of XML.
BOOKS
2. Scott Guelich, Shishir Gundavaram, Gunther Birzniek; CGI Programming with Perl 2/e, O’Reilly
3. Doug Tidwell, James Snell, Pavel Kulchenko; Programming Web Services with SOAP, O’Reilly
5. Yong, XML Step by Step, PHI.
Program Design & File Structures
IT-209

L     T
3     1

Theory: 60
Sessional: 40

1. **Introduction to File Systems**
Characteristics and working of magnetic tapes, magnetic disks, optical disks; File organizations, file operations, file systems, file directories, device control, channel and buffer management, Sequential File Organization – Creating, retrieving and updating sequential files & their performance; merging – natural, balanced, polyphase, cascade and their performance; relative file organization – definition, addressing techniques, problem of collision and its solutions, performance; indexed sequential file organization – definition, application, implementation strategies; multi-key file organization – multi-key access, inverted file organization, multi-list file organization.

2. **File Organizations & Indexes**
cost model, comparison of 3 file organizations, overview & properties of indexes; Tree-structured Indexes- ISAM, B+ trees: format & operations, B+ trees in practice; Hash based Indexing- static hashing, extendible hashing, linear hashing, comparisons; Using B+ trees for sorting.

**BOOKS**
List of Practical

1. To study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, & EX-NOR).
2. To design and verify a half and full adder circuits.
3. To design a 4 bit adder/subtract using IC 7483.
4. To design and implement a 4:1 multiplexer.
5. To design and implement a 1:4 demultiplexer.
6. Verify the truth table of a 4-bit comparator using IC 7485.
7. To design and verify a 2:4 decoder.
8. To design and implement a 2:4 encoder.
9. To verify the operation of a D and JK flip-flop using ICs 7474 AND 7473.
10. To design and verify the operation of RS, T, D, and JK flip-flops using logic gates.
11. To verify the operation of a Mod-10 counter.
12. To design and implement the operation of a Mod-16 counter using JK flip-flops
13. To design and implement a Mod-10 counter using JK flip flops and logic gates.
14. To verify the operation of a 4 bit shift register using IC 7495.
15. To design and verify the operation of a 4-bit shift left register using D flip-flops.
16. To design and verify the operation of a 4-bit shift right register using D flip-flop.
Object Oriented Programming using C++ (P)

IT-213

List of Practical

1. (a) Model a geometric point to find distance between two points.
   (b) Model complex numbers and their operations.

2. Describe a class called TOLL-BOOTH with the following data items
   unsigned int - to hold the number of cars passing through the booth,
   double - to hold the total amount collected.
   Include the following member functions:
   * a constructor that sets both the data fields to zero.
   * PAYINGCAR( ) that increases the numbers of cars by one and increase
     the total amount by 2.50.
   * NOPAYING( ) that increases the number of cars but keeps the total amount unchanged.
   * DISPLAY( ) that displays both the total number of cars passing and the total number of
     amount collected.

   Write main( ) to test the class thoroughly.

3. Create a class rational which represents a numerical value by two double
   values- NUMERATOR and DENOMINATOR . Include the following public member
   functions:
   * constructor with no arguments (default)
   * constructor with two arguments.
   * void reduce () that reduces the rational number by eliminating the highest common factor
     between the numerator and denominator.
   * overload + operator to add two rational numbers.
   * overload >> operator to enable input through cin.
   * overload << operator to enable output through cout.

   Write a main () to test all the functions in the class .

4. Consider the following class definition
   class father {
   
}
protected : int age;

public;
    father (int x){age =x;}
    virtual void iam()
        {cout <<"I AM THE FATHER, my age is:"<<age<<end1;}
};

Derive the two classes son and daughter from the above class and for each, define iam() to write out similar but appropriate messages. You should also define suitable constructors for these classes.

Now, write a main () that creates objects of the three classes and then calls iam() for them. Declare pointer to father. Successively, assign addresses of objects of the two derived classes to this pointer and in each case, call iam() through the pointer to demonstrate polymorphism in action.

5. A thermostat is a device that keeps a system at a constants temperature. It behaves like a temperature gauge that is capable of getting the current temperature from the system. It is also a switch that can be turned "on" and "off". The thermostat monitors the temp. in the following manner:
    if the current temp. falls below 95% of the required temp., it turns itself "on". On the other hand , if the current temp. exceeds 1.05 of the required temp. ,it turns itself "off".In all other cases ,its on-off status remain un changed.

Implement classes for temp. gauge and switch(named switch) with suitable data and member functions. The temp. gauge class must have a member function get_temp() that will pretend to get the current temp. of the system by actually reading it from the keyboard.

Now, implement thermostat class in both the following ways:
    a) Develop a class called thermostat that include objects of temp. gauge and switch as its member (aggregation).
    b) Develop a class called thermostat that inherits the data functions of temp. gauge and switch(multiple inheritance).
Write main ( ) to test all the features of above-mentioned classes.

6. Write a program that creates a binary file  by reading the data for the students  from the terminal. The data of each student consist of roll no., name (a string of 30 or lesser no. of characters) and marks.

7. Using the file created in problem 6, write a program to display the roll no. and names of the students who have passed (has obtained 50 or more).

8. You are to create a file containing n records. Each record relates to a historical event and the year in which the event took place
    Some examples are:
        India Wins Freedom 1947
        Amartya Sen Gets Nobel 1998
        First World War Begins 1914
    The data should be read from terminal while creating the file.
9. A hospital wants to create a database regarding its indoor patients. The information to store include
   (a) Name of the patient
   (b) Date of admission
   (c) Disease
   (d) Date of discharge
   Create a structure to store the date (year, month and date as its members). Create a base class to store
   the above information. The member function should include functions to enter information and display a
   list of all the patients in the databases. Create a derived class to store the age of the patients. List the
   information about all the pediatric patients (less than twelve years in age).

10. Define a class to store the time at a point. The data members should include hr., min., and sec. to store
    hours, minutes and seconds. The member functions should include functions for reading the time and
    displaying the same. Add a friend function to add two times. Write a program, using the above declaration,
    to read two times and add them.

11. Write a program to read two matrices and find their product. Use operator overloading so that
    the statement for multiplying the matrices may be written as $Z = X \cdot Y$ where $X$, $Y$ and $Z$
    are matrices.

12. Write a program to read a number and display its square, square root, cube and cube root.
    Use a virtual function to display any one of the above.

13. Make a class Employee with a name and salary. Make a class Manager inherit from Employee.
    Add an instance variable, named department, of type String. Supply a method to toString that prints
    the manager’s name, department and salary. Make a class Executive inherit from Manager. Supply a
    method toString that prints the string “Executive” followed by the information stored in the Manager
    superclass object. Supply a test program that tests these classes and functions.

14. Write a superclass Worker and subclass HourlyWorker and SalariedWorker. Every
    worker has a name and a salary rate. Write a method computePay(int hours) that computes
    the weekly pay for every worker. An hourly worker gets paid the hourly wage for the actual
    number of hours worked, if hours is at most 40. If the hourly worker worked more than 40
    hours, the excess is paid at time and a half. The salaried worker gets paid the hourly wage for
    40 hours, no matter what the actual number of hours is. Write a static method that uses
    polymorphism to compute the pay of any Worker. Supply a test program that tests these
    classes and functions.
List of Practical

1. Define two strings as arrays. Read them using %s. Using pointers, concatenate them without using string.h.
2. Define a pointer to an integer; read a list of n numbers using dynamic memory allocation and find average of these numbers.
3. Create a file containing 26 alphabets (A to Z) in separate lines.
4. Copy a file to another. Source file name and destination file name are input from the user.
5. Write a program for binary search (successful and unsuccessful both).
6. Sort n numbers using quick/merge/selection sort. Also count the number of exchanges in each case.
7. Write a program for expression evaluation using stacks.
8. Write a program for infix to postfix conversion.
9. Create a singly linked list and reverse it in the same list.
10. Write a program for a doubly linked list giving following option, insertion, deletion, retrieval,
11. Write a program to implement queues using linked list with option; list of elements in queue, insertion, and deletion.
12. Write a program to implement stacks using linked list with options push and pop.
13. Write a program for multiplication of two polynomials using linked list.
14. Write a program to implement binary trees. Depending on the choice, inorder/ preorder/ postorder traversal is done.
15. Implement heap sort. Show the contents of heap after each adjustment of element i.e. n outputs should be printed if list has n elements.
List of Practical

1. Chalk out the storyboard and design of Dairy Food Limited. As the name reflects your site provides dairy products and aims at opening an online store. Your storyboard should cover all the features that you plan to have on the site.
2. Create your own page with your favorite hobbies.
3. Create a Menu or a Table of content web page. Each menu item or section of the table of content should load a different web page. For example, if the user clicks on Menu one or section 1 then the link should take him to respective menu html. Or section and so on.
4. Create a web site for your College,
5. Create a frameset that is divided into three sections. The frameset should have three zones.
   - the topmost section of the frameset should take up about just 15% of the browser window. Name this frame title.
   - The middle section should be 70% of the browser window. Name this frame title.
   - The lower most sections should also be about 15% of the browser window. Name this section as menu. Create pages for each section. For the lowermost section, create page that loads the content into the middle section. The topmost section should contain a page describing the web page itself.
6. Create a web page, which displays the map of your Country Link, each city/state on the image using image map, such that the respective HTML page of the city/state is displayed when the user selects an area.
7. Add the tickertape applet to your page by customizing it for the following settings:
   - Increase the count by one.
   - Accordingly update the message count.
   - Change the text color to (237,192,171)
   - Experiment with changing the scrolling speed.
   - Customize the message text as per your page requirement.
8. Incorporate a quest book into the Dairy Food Webpage and use Java Script to build validations into the form.
9. Use Style sheet to modify the following:
   - Change background to modify the following.
   - Change font, type, face and color.
   - Align Text.
   - Remove underlines from hyperlinks.
10. Use any Web Server to set up your website.
Program Design and File Structures (P)

IT-219

L      T      P                  Practical:    40
-      -      2                  Sessional:    60

Note: Implement following programs in C language.

1. Modular program development of a simple text based calculator.
2. Modify above design to develop scientific calculator.
3. Computation of \(^nC_m\) using Recursion.
4. Generate Fibonacci series using recursion.
5. Implement natural merge and polyphase merge.
6. Implement a GUI/Mouse driven simple calculator.