University of Pune

MCA - I

(Under Science Faculty)

Lab Course – 206

Lab Manual

Name: ____

Roll No. ______________ Academic Year: 20_____ - 20_____

College Name: ____

Preface:

In pursuance of the decision to implement credit system at the post graduate level and ensure continuous assessment, the UOP has decided to implement the credit & semester system (CSS) in all its affiliated college and recognized institutions where post graduate courses are conducted.

Assessment shall consist of continuous assessment (CA and END of Semester Examination (ESE)). Each shall have an equal weight age of 50%.

Assessment and grade point average

- The system of evaluation is as follows: Each CA and ESE will be evaluated in terms of marks. The marks for CA and ESE will be added together and then converted into a grade and later a grade point average.
- Results will be declared for each semester.
- After the gain of minimum number of credits toads a completion of a PG program, a student will get a grade sheet with total grades earned and a grade point average.
- Marks /Grade/Grade point

Marks	**Grade**	**Grade Point**
100 to 75	O : Outstanding	06
74 to 65	A : Very good	05
64 to 55	B : Good	04
54 to 50	C : Average	03
49 to 45	D : Satisfactory	02
44 to 40	E : Pass	01
39 to 0	F : Fail	00

Final Grade Points	
Grade Points	**Final Grade**
5.00 – 6.00	O
4.50 – 4.99	A
3.50 – 4.49	B
2.50 – 3.49	C
1.50 – 2.49	D
0.50 – 1.49	E
0.00 – 0.49	F

Practical Evaluation Format

- The internal continuous assessment will be 50M and end semester lab examination will be of 50M
- The outline of distribution of Practical Marks for various aspects /mechanisms towards ***Continuous Assessment*** is as follows:

Sr. No.	Distribution Marks	Marks
1	**Journal** • After finishing every practical instructor should give the marks to respective assignment and evaluate those marks at the end of the semester out of 15	**15**
2.	**Database** • Cast Study assignment.	**15**
5.	**Internal Practical Evaluation** • Respective subject teachers should evaluate Data Structures and Object Oriented Programming using C++ assignments for 10M each by conducting mock tests	**20**
	Total	**50M**

- The outline of distribution of Practical Marks for towards ***End Semester lab examination*** is as follows:

Sr. No.	Distribution Marks	Marks
1	**Data Structures program**	**20**
2.	**Object Oriented Programming using C++ program**	**20**
3.	**Viva**	**5**
4.	**Database Case Study Evaluation**	**5**
	Total	**50M**

Introduction

1. About the work book

This workbook is intended to be used by MCA-I (Computer Science) students for the laboratory course CA 206. In MCA, hands-on laboratory experience is critical to the understanding of theoretical concepts studied in the theory courses. This workbook provides numerous computing problems covering all difficulty levels.

The objectives of this book are

1) Defining clearly the scope of the course
2) Bringing uniformity in the way the course is conducted across different colleges
3) Continuous assessment of the course
4) Bring in variation and variety in the experiments carried out by different students in a batch
5) Catering to the need of slow paced as well as fast paced learners

2. How to use this workbook

This workbook is mandatory for the completion of the laboratory course. It is a measure of the performance of the student in the laboratory for the entire duration of the course.

2.1. Instructions to the students

1) Students are expected to carry this book every time they come to the lab for practical's
2) Student should maintain separate journal for the source code
3) Student should read the topics mentioned in Reading section of this book **before** coming for practical
4) Students should solve only those exercises which are selected by instructor as a part of journal activity. However, students are free to solve additional exercises to do more practice for their practical examination

Exercise	Difficult Level	Rule
Set A	Easy	All programs are compulsory
Set B	Medium	At least 1 program is compulsory
Set C	Hard	Not compulsory

5) Student will be assessed for each exercise on a scale of 5

Not Done	0
Incomplete	1
Late Complete	2
Needs Improvement	3
Complete	4
Well Done	5

2.2. Instruction to the Instructors

1) Explain the assignment and related concepts in around ten minutes using white board if required or by demonstrating the software
2) Choose appropriate problems to be solved by student
3) Make sure that students follow the instruction as given above

4) After a student completes a specific set, the instructor has to verify the outputs and sign in the provided space after the activity
5) Ensure that the students use good programming practices
6) You should evaluate each assignment carried out by a student on a scale of 5 as specified above by ticking appropriate box
7) The value should also be entered on assignment completion page

2.3. Instructions to the Lab administrator

You have to ensure appropriate hardware and software is made available to each student.

The operating system and software requirements on server side and also client side are as given below

1) Server Side (Operating System)
    a. * Fedora Core Linux
    b. * Microsoft Windows Server 2003
    c. Servers Side (software's to be installed)
       In Linux – C, C++, awk, shell, perl, postgresql/Mysql
       In WinXP -- MSOffice

2) Client Side (Operating System)
    a. * Red Hat Linux and Fedora Core
    b. * Microsoft Windows XP
    c. Client Side (software's to be installed)
       In Linux – C, C++, awk, shell, perl, postgresql/mysql
       In WinXP -- MSOffice

Compiled From : S.Y.B.Sc (computer science Lab book)

Compiled by :

Ms. Priya Wagh, Nowrosjee Wadia College, Pune
Ms. Arati Tilokani, Nowrosjee Wadia College, Pune
Ms. Nutan Borse, Nowrosjee Wadia College, Pune

Reviewed by:

Ms. Poonam Ponde, Nowrosjee Wadia College, Pune

SECTION I - Assignment Evaluation

Sr. No	Assignment Name	Teachers Sign	Assignment Marks	Viva Marks
1	Sorting Techniques			
2	Searching Techniques			
3	Linked List			
4	Stack			
5	Queue			
6	Tree			
7	Graph			
TOTAL				

SECTION II - Assignment Evaluation

Sr. No	Assignment Name	Teachers Sign	Assignment Marks	Viva Marks
1	Introduction to C++ Programming			
2	Functions In C++			
3	Classes and Objects			
4	Constructors and Destructors			
5	Operator Overloading			
6	Inheritance			
7	Formatted I/O			
8	Exception Handling			
9	File I/O			
10	Templates			
TOTAL				

Name Of Batch Incharge:

Signature Of Batch Incharge: **Head of Department**

Date:

SECTION I
DATA STRUCTURES USING ‘C’

SESSION 1: Sorting Techniques

Practical Assignments

SETA

1. Write a C program to read n numbers from the user and sort them in ascending and descending order using Bubble sort method. Use step count, swap count, comp count in your program to calculate time complexity and at end display the 3 counts

2. Write a C program to read n numbers from the user and sort them in ascending and descending order using Insertion sort method. Use step count, swap count, comp count in your program to calculate time complexity.

SET B

1. Write a C program to read n numbers from the user and sort them in ascending, descending order using Merge sort. Use step count, swap count, comp count in your program to calculate time complexity.

2. Write a C program to read n numbers from the user and sort them in ascending descending order using Quick sort. Use step count, swap count, comp count in your program to calculate time complexity.

SETC

1. Write a program that accept name of persons into array and sort them in alphabetical order.

2. Write a program that accept employee name, age and salary into array and sort them in descending order of salary. [Hint: Use array of structure]

Assignment Evaluation

0: Not Done[ ]   1: Incomplete[ ]   2.LateComplete[ ]

3: Needs Improvement[ ]   4: Complete[ ]   5:WellDone[ ]

Signature of the Instructor   Date of Completion

SESSION 2: Searching Techniques

Start Date //

Practical Assignments

SET A

1. Write a program to accept list of n numbers and search a given number using Linear Search. Use stepcnt, compcnt in your program to calculate time complexity and at end display the3 counts

2. Write a program to accept list of cities and search a given city using Linear Search. Use stepcnt, compcnt in your program to calculate time complexity and at end display the3 counts

SETB

1. Write a program to accept list of sorted n numbers and search a given number using Binary Search.

2. Write a program to accept list of student names in alphabetical order and search a given name using Binary Search.

SETC

Accept city name, area and population for n cities into array. Find the area and population of given city using

1. Linear Search method
2. Binary Search method

(Hint: Use array of structures and sort the array on city name before applying binary search)

Assignment Evaluation

0:Not Done[ ] 1:Incomplete[ ] 2.LateComplete[ ]
3:NeedsImprovement[ ] 4:Complete[ ] 5:WellDone[ ]

Signature of the Instructor Date of Completion

SESSION 3: Linked List

Start Date / /

Practical Assignments

SET A

1.Write a C program to implement singly linked list of integers.

2.Write a C program to implement circular singly linked list.

3. Write a C program to implement doubly circular linked list.

SETB

1. Write a C program to perform union, intersection operations on two sorted linked lists.

2. Write a C program to add two polynomials using linked list

2. Write a C program to merge two sorted linked list.

SETC

1. Write a C program to sort the singly linked list of integers. (Create an unsorted list and then sort it.

3. Write a C program to create linked list of strings. Display all strings containing same substring.(Accept substring from user.)

Assignment Evaluation

0:Not Done[ ] 1:Incomplete[ ] 2.LateComplete[ ]
3:NeedsImprovement[ ] 4:Complete[ ] 5:WellDone[ ]

Signature of the Instructor Date of Completion

SESSION 4: Stack

Start Date / /

Practical Assignments

SET A

1. Write a C program to reverse a string using static stack. (Create Stack.h file.)

2. Write a C program to check whether the string is palindrome or not using static stack.

3. Write a C program to evaluate a postfix expression.

SETB

1. Write a C program to convert infix expression into postfix form

2. Write a C program to implement a stack of integers using linked list. The program should be menu driven with the following options: PUSH, POP

SETC

1.Write a C program to simulate recursion of following functions:

a. Factorial of n numbers

b. Power function

Assignment Evaluation

0:Not Done[ ] 1:Incomplete[ ] 2.LateComplete[ ]

3:NeedsImprovement[ ] 4:Complete[ ] 5:WellDone[ ]

Signature of the Instructor Date of Completion

SESSION 5: Queue

Start Date / /

Practical Assignments

SET A

1. Write a C program to implement linear queue of strings using array. Store the structure and queue functions in ƒqueue.h, and write the main function in a .C file. The program should be menu driven with the options : ADD, DELETE, EXIT.
2. Write a C program to implement circular queue of integers using array.

SETB

1.Write a C program to implement static ascending priority queue.

2.Write a C program to implement a queue of integers using linked list. The program should be menu driven with the following options: ADD,DELETE,EXIT.

SETC

1. Write a C program to implement static DEQUE.

2. Write a C program to implement Multiple Queues using single array.

Assignment Evaluation

0:Not Done[ ] 1:Incomplete[ ] 2.LateComplete[ ]

3:NeedsImprovement[ ] 4:Complete[ ] 5:WellDone[ ]

Signature of the Instructor Date of Completion

SESSION 6: Tree

Start Date / /

Practical Assignments

SET A

1. a) Write a menu driven program for Binary Search Tree creation and inorder, preorder, post order recursive traversal of all nodes.
   b. Modify the above program for performing following operations:
      a. Search a value.
      b. Insertion of a value

SETB

1. Write a program to create Binary Search tree with the following functions:
   a) tcopy _ function to make an identical copy of a tree T, returns root of newly created tree
   b) tmirror - returns the mirror of a tree
   c) tleafcnt, tnodes _counting leaf nodes and total nodes of a tree T
2. Write a program to sort n integers using Heap Sort

SETC

1. Write a program to perform Non_recursive traversal using stack(inorder, preorder, postorder) on the BST
3. Write a function to delete a node from a Binary Search tree
4. Write a program to create height balanced tree. Write appropriate functions for different cases_ LL, RR, LR, RL to rotate the tree and maintain thebalance.

Assignment Evaluation

0:Not Done[ ] 1:Incomplete[ ] 2.LateComplete[ ]

3:NeedsImprovement[ ] 4:Complete[ ] 5:WellDone[ ]

Signature of the Instructor Date of Completion

SESSION 7: Graph

Start Date / /

Practical Assignments

SET A

1. Write a program to read a graph as adjacency matrix. Calculate indegree, outdegree and total degree of each vertex.
2. Write a program to implement BFS and DFS.(Use recursive DFS function.)
3. Write a program to read a graph as adjacency matrix and convert it into adjacency list.

SET B

1. Write a program to implement graph as a multi-list .

Assignment Evaluation

0:Not Done[ ] 1:Incomplete[ ] 2.LateComplete[ ]

3:NeedsImprovement[ ] 4:Complete[ ] 5:WellDone[ ]

Signature of the Instructor Date of Completion

End of Session

SECTION II
OBJECT ORIENTED PROGRAMMING
USING C++

Exercise 1

No. Of Slots: 1-2

Introduction to C++ Programming

Reading

You should read the following topics before starting this exercise

- ➢ Difference between C and C++
- ➢ Features of C++
- ➢ C++ functions
- ➢ Reference variables
- ➢ Call by reference
- ➢ Return by reference

Practical Assignment

SET A

1. Write a program to compute the area and perimeter of a rectangle 3 inches wide by 5 inches long. What changes must be made to the program so it works for a rectangle 6.8 inches wide by 2.3 inches long?
2. Write a program to perform date arithmetic, such as how many days there are between 6/1/90 and 8/3/92.
3. Write a program to calculate the monthly telephone bills as per the following rule:
   Minimum Rs. 200 for upto 100 calls.
   Plus Rs. 0.60 per call for next 50 calls.
   Plus Rs. 0.50 per call for next 50 calls.
   Plus Rs. 0.40 per call for any call beyond 200 calls.
4. Write a program that prints the sizes of the fundamental types, a few pointer types, and a few enumerations of your choice. Use the *sizeof* operator.

SET B

1. Write a program in C++ using switch statement that contain option as under
   Enter 1--> To Find Largest Number Among Three Variables.
   Enter 2--> To Find ODD or EVEN
   Enter 3--> To Find Condition of Water
   Enter 4--> To Find Grade Of Student
2. Get month, day & year from user in main(). Call the function printDate(int, int, int), Put a check in printDate() using if statement (month < 1 || month > 12 || day < 1 || day > 31 || year < 0) & if it violates the rule display *—Must Enter a Valid Date",* Using switch Statement get the month, Day and year are displayed normally, Termination should be on entering 0 in months

SET C

1. Write a C++ program to implement the Number Guessing Game. In this game the computer chooses a random number between 1 and 100, and the player tries to guess the number in as few attempts as possible. Each time the player enters a guess, the computer tells him whether the guess is too high, too low, or right. Once the player guesses the number, the game is over.
2. Develop a code in C++ that generate the following series .Use nested while loop!

Series No.1

1
1 2
1 2 3
1 2 3 4

Series No.2

1
2 2
3 3 3
4 4 4 4

Series No.3

1
2 3
4 5 6
7 8 9 10

Assignment Evaluation

0: Not Done [ ]   1: Incomplete [ ]   2.Late Complete [ ]
3: Needs Improvement [ ]   4: Complete [ ]   5: Well Done [ ]

Signature of the Instructor   **Date of Completion ____/____/______**

Exercise 2

No. Of Slots: 2

Functions in C++

Reading

You should read the following topics before starting this exercise

- Concept of function overloading in C++
- Default arguments, const arguments in C++
- Inline Functions, Call by reference

Practical Assignment

SET A

1. Write a function called zero_small() that has two integer and sets the smaller of the two numbers to 0. Display the result in main. (Use reference variables)
2. Write a function in C++ to calculate simple interest for a given amount and period. If the user does not provide rate and number of years, use a default value of 10% and 2 years.
3. Write a C++ program to write a function called as power() to raise a number m to a power n. The function takes double value of for m and int value for n, and returns the result correctly. Use a default value of 2 for n to make the function to calculate squares when this argument is omitted. Now write one more function with the same name "power" but which takes int value of m. Call both these functions in main by passing user given values.

SET B

1. Write a C++ functions to sort n elements. The default sorting is done in ascending order. If sorting is required in the descending order, an additional argument of 1 is passed to the function. Use any sorting technique. Function overloading is expected.
2. Write a program to calculate area of circle, rectangle and cylinder. Overload function area for each.
3. Write a C++ program to accept records of n students using structure and store it in an array (roll number, name, percentage). Overload the following search functions:
   int search(student s[], int n, int roll);
   int search(student s[], int n, char Name[]);
   int search(student s[],int n, float p);

SET C

1. Write a C++ function which converts a decimal number to another base. The function takes the decimal number as one argument and the base as another. If the second argument passed is 'h', convert it to hex, 'o' for octal and if no argument is passed, convert it to binary.

Assignment Evaluation

0: Not Done [ ] 1: Incomplete [ ] 2.Late Complete [ ]
3: Needs Improvement [ ] 4: Complete [ ] 5: Well Done [ ]

Signature of the Instructor **Date of Completion** ____/____/______

Exercise 3

No. Of Slots: **2**

Classes & Objects

Reading

You should read following topics before starting this exercise:

- Concept of Class in C++
- Types of Member Functions in Class
- Creating Objects
- Array of Objects
- Access Specifiers
- Static Class Member Functions
- Friend Class and Friend Functions
- **const** Member Functions

Practical Assignment

SET A

1. Define a class in C++ with following description:
   Private Members
   A data member Flight number of type integer
   A data member Destination of type string
   A data member Distance of type float
   A data member Fuel of type float
   A member function CALFUEL() to calculate the value of Fuel as per the following criteria

Distance	Fuel
<=1000	500
more than 1000 and <=2000	1100
more than 2000	2200

   Public Members
   A function FEEDINFO() to allow user to enter values for Flight Number, Destination, Distance & call function CALFUEL() to calculate the quantity of Fuel
   A function SHOWINFO() to allow user to view the content of all the data members
2. Define a class REPORT with the following specification:
   Private members :

adno	4 digit admission number
name	20 characters
marks	an array of 5 floating point values
average	average marks obtained
GETAVG()	a function to compute the average obtained in five subject

   Public members:
   READINFO() function to accept values for adno, name, marks. Invoke the function GETAVG()
   DISPLAYINFO() function to display all data members of report on the screen.
   You should give function definitions.
3. Design an airline reservation data structure that contains the following data:

- ➢ Flight number
- ➢ Originating airport code (3 characters)
- ➢ Destination airport code (3 characters)
- ➢ Departure time
- ➢ Arrival time

Write a program that lists all planes leaving from two airports specified by the user.

4. Write necessary class and member function definitions for a cricket player object. The program should accept the details from user (maximum 10)
   The details of the player are player code, name, runs, innings played and number of times not out.
   The program should contain following menu
   - ✓ Enter details of players
   - ✓ Display average runs of a single player
   - ✓ Average runs of all players
   - ✓ Display the list of players in sorted order as per runs

   (Use function overloading)
5. Create two classes dist1(meters, centimetres) and dist2(feet, inches). Accept two distances from the user, one in meters and centimetres and the other in feet and inches. Find the sum and difference in between the two distances. Display the result in both, meters and centimetres as well as feet and inches (use friend function).

SET B

1. Write a class to represent a vector. Include member functions to perform the following tasks:
   - ✓ To create the vector
   - ✓ To modify the value of given element
   - ✓ To multiply by a scalar value
   - ✓ To display the vector in the form(10,20,30,....)
2. Define a class to represent a bank account. Include the following members:
   Data members
   I. Name of the depositor
   II. Account number
   III. Type of account
   IV. Balance amount in the account.

   Member functions:
   I. To assign initial values
   II. To deposit an amount
   III. To withdraw an amount after checking the balance
   IV. To display name and balance.

   Write a main program to test the program.

SET C

1. Write the definition for a class called **Rectangle** that has floating point data members length and width. The class has the following member functions:
   - ➢ **void setlength(float)** to set the length data member
   - ➢ **void setwidth(float)** to set the width data member
   - ➢ **float perimeter()** to calculate and return the perimeter of the rectangle
   - ➢ **float area()** to calculate and return the area of the rectangle

- ➢ **void show()** to display the length and width of the rectangle
- ➢ **int sameArea(Rectangle)** that has one parameter of type Rectangle. sameArea returns 1 if the two Rectangles have the same area, and returns 0 if they don't.

A. Write the definitions for each of the above member functions.

B. Write main function to create two rectangle objects. Set the length and width of the first rectangle to 5 and 2.5. Set the length and width of the second rectangle to 5 and display each rectangle and its area and perimeter.

C. Check whether the two Rectangles have the same area and print a message indicating the result. Set the length and width of the first rectangle to 15 and 6.3. Display each Rectangle and its area and perimeter again. Again, check whether the two Rectangles have the same area and print a message indicating the result.

2. Write the definition for a class called **time** that has hours and minutes as integer. The class has the following member functions:
   - ➢ **void settime(int, int)** to set the specified value in object
   - ➢ **void showtime()** to display time object
   - ➢ **time sum(time)** to sum two time object & return time

   A. Write the definitions for each of the above member functions.

   B. Write main function to create three time objects. Set the value in two objects and call sum() to calculate sum and assign it in third object.

   C. Display all time objects.

3. Define a class named Sequence for storing sorted strings. Following are the member functions for Sequence class:
   - ✓ Insert which inserts a new string into its sort position.
   - ✓ Delete which deletes an existing string.
   - ✓ Find which searches the sequence for a given string and returns true if it finds it, and false otherwise.
   - ✓ Print which prints the sequence strings.

Assignment Evaluation

0: Not Done [ ]　　1: Incomplete [ ]　　2.Late Complete [ ]

3: Needs Improvement [ ]　　4: Complete [ ]　　5: Well Done [ ]

Signature of the Instructor　　**Date of Completion** ____/____/______

Exercise 4

No. Of Slots: 2

Constructors & Destructors

Reading

You should read following topics before starting this exercise:

- What is constructor?
- Role of constructor in class
- Types of constructor
- Role of destructor and how it is different from constructor?

Practical Assignment

SET A

1. Define a class named Complex for representing complex numbers. A complex number has the general form a + *i*b, where a is the real part and b is the imaginary part (*i* stands for imaginary). Complex arithmetic rules are as follows:
   (a + *i*b) + (c + *i*d) = (a + c) + *i*(b + d)
   (a + *i*b) – (c + *i*d) = (a + c) – *i*(b + d)
   (a + *i*b) * (c + *i*d) = (ac – bd) + *i*(bc + ad)
   Define the class having data members real and imaginary. Define appropriate constructors for the class. Perform the above operations on class objects.
2. Define a class Fraction having members numerator and denominator. Define parameterized and default constructors (default values 0 and 1). Create two fraction objects and perform the following operations by defining member functions: Addition, Subtraction, Multiplication and division.

SET B

1. Write a c++ program to read the information like plant_name, plant_code, plant_type, price. Construct the database with suitable member functions for initializing and for destroying the data, viz. constructor, copy constructor, destructor.

SET C

1. A common place to buy candy is from a machine. The machine sells candies, chips, gum, and cookies. You have been asked to write a program for this candy machine. The program should do the following:
   1. Show the customer the different products sold by the candy machine.
   2. Let the customer make the selection.
   3. Show the customer the cost of the item selected.
   4. Accept money from the customer.
   5. Release the item.
2. The machine has two main components: a built-in cash register and several dispensers to hold and release the products.
3. Define class cashRegister in C++ with the following descriptions :
   Private Members:
   cashOnHand of type integer
   Public Members:

A default constructor cashRegister() sets the cash in the register to 500.

A constructor cashRegister(int) sets the cash in the register to a specific amount.

A function getCurrentBalance() which returns value of cashOnHand

A function acceptAmount(int) to receive the amount deposited by the customer and update the amount in the register

4. Define class dispenserType in C++ with the following descriptions :

Private Members:

numberOfItems of type integer

cost of type integer

Public Members:

A default constructor dispenserType () sets the cost and number of items in the dispenser to 50 each.

A constructor dispenserType (int,int) sets the cost and number of items in the dispenser to the values specified by the user.

A function getNoOfItems() to return the value of numberOfItems.

A function getCost() to return the value of cost.

A function makeSale() to reduce the number of items by 1.

5. When the program executes, it must do the following:

1. Show the different products sold by the candy machine.

2. Show how to select a particular product.

Once the user has made the appropriate selection, the candy machine must act accordingly. If the user has opted to buy a product and that product is available, the candy machine should show the cost of the product and ask the user to deposit the money. If the amount deposited is at least the cost of the item, the candy machine should sell the item and display an appropriate message.

Divide this program into three functions: showSelection, sellProduct, and main.

The function sellProduct must have access to the dispenser holding the product (to decrement the number of items in the dispenser by 1 and to show the cost of the item) as well as the cash register (to update the cash). Therefore, this function has two parameters: one corresponding to the dispenser and the other corresponding to the cash register.

Assignment Evaluation

0: Not Done [ ]   1: Incomplete [ ]   2.Late Complete [ ]
3: Needs Improvement [ ]   4: Complete [ ]   5: Well Done [ ]

Signature of the Instructor   **Date of Completion ____/____/______**

Exercise 5

No. Of Slots: 2

Operator Overloading

Reading

You should read following topics before starting the exercise:

- Operator Overloading and polymorphism in C++
- Operators in C++ and functions associated with each operator
- Overloading unary/binary operators as member function and friend function
- Rules for operator overloading
- insertion (>>) and extraction(<<) operators
- Concept of **this** pointer in C+

Practical Assignment

SET A

1. Write a C++ program to overload ! operator to find factorial of an INTEGER object.
2. Create a class **Rational** to represent a **Rational** number. **Perform the Basic Arithmetic operations: Addition, Subtraction, Multiplication and Division for Two Rational Numbers.**
3. Create a class **String** to represent a string. Overload the + operator to concatenate 2 strings, <=, == , >= to compare 2 strings

SET B

1. Define a class Date. Overload the operators << and >> for this class to shift a date by specific number of days.
2. Create a class matrix which stores a matrix of integers of given size.
   Write a necessary member functions. Overload the following operators
   + Adds two matrices and stores in the third.
   == Returns 1 if the two matrices are same otherwise 0.
   (Use dynamic memory allocation)
3. Create a class Fraction having two data members (numerator, denominator). The default values should be 0 and 1. Overload the following operators
   << Insertion
   >> Extraction (display in reduced form)
   < Returns 1 if first fraction is less than second
   Accept n fraction objects and sort them in the ascending order.

SET C

1. Write a program to overload the new and delete operator
2. Sparse matrices are used in a number of numerical methods (e.g., finite element analysis). A sparse matrix is one which has the great majority of its elements set to zero. In practice, sparse matrices of sizes up to 500 □ 500 are not uncommon. On a machine which uses a 64-bit representation for reals, storing such a matrix as an array would require 2 megabytes of storage. A more economic representation would record only nonzero elements together with their positions in the matrix. Define a SparseMatrix class which uses a linked-list to record only nonzero elements, and

overload the +, -, and * operators for it. Also define an appropriate memberwise initialization constructor and memberwise assignment operator for the class.

Assignment Evaluation

0: Not Done [ ] 1: Incomplete [ ] 2.Late Complete [ ]
3: Needs Improvement [ ] 4: Complete [ ] 5: Well Done [ ]

Signature of the Instructor **Date of Completion** ____/____/______

Exercise 6 No. Of Slots: 3
Inheritance

Reading

You should read following topics before starting this exercise:

- Concept of inheritance.
- Types of inheritance
- Polymorphism – run time and compile time
- Virtual inheritance
- Virtual base classes
- Abstract class
- Virtual and pure virtual functions
- Concept of Overriding - virtual functions and pure virtual functions

Practical Assignment

Set A

1. Create base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive three specific classes called triangle, circle and rectangle from the base shape. Add to the base class, a member function get_data() to initialize base class data members and display_area() to compute and display area. Make display_area() as a virtual function.(Hint: **use default value for one parameter while accepting values for shape circle.)
2. Write a program in C++ to read the following information from the keyboard in which the base class consists of employee name, code and designation. The derived class Manager which contains the data members namely, year of experience and salary. Display the whole information on the screen

Set B

1. Design a base class person(name,address,phone_no).Derive a class employee(e_no,ename) from person.Derive a class manager(designation,department,basic_salary) from employee.
   Write a menu driven program to
   - ✓ Accept all details of 'n' manager.
   - ✓ Display manager having highest salary.

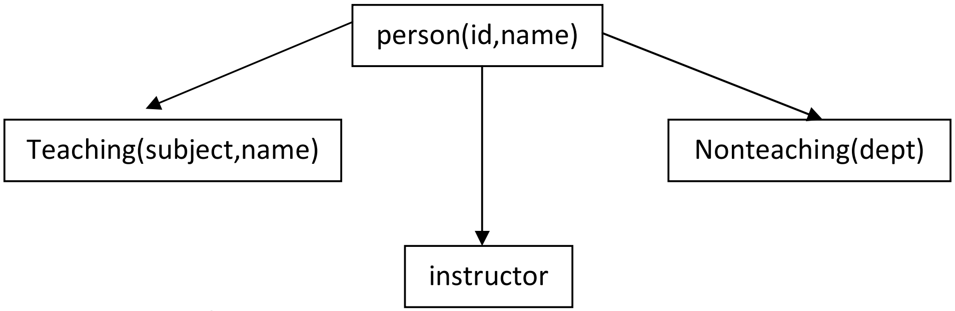
   (Use private inheritance).
2. A company markets books audio tapes. Prepare a class publication having members(title,price).From this class derive two classes
   - ✓ Book which adds a page count.
   - ✓ Tape which adds a playing time in minutes.

   The program should contain following menu.
   - ✓ Display all books.
   - ✓ Display all tapes.
   - ✓ All books having pages>n.
   - ✓ All tapes having palying time>=n minutes.

3. Design two base classes Student(roll_no,name,class) and Marks(mark[3]). Design one derived class Result(percentage,grade).Write a menu driven program with following options
   - ✓ Add the details of ‘n’ students with marks.
   - ✓ Display result of all students in descending order of percentage.

Set C

1. Design the classes using following hierarchical inheritance teaching



Each class has member functions accept and display. Write a program to accept details of n instructors and display the details.

Assignment Evaluation

0: Not Done [ ]   1: Incomplete [ ]   2.Late Complete [ ]
3: Needs Improvement [ ]   4: Complete [ ]   5: Well Done [ ]

Signature of the Instructor   **Date of Completion ____/____/______**

<u>Exercise 7</u>

No. Of Slots: **1**

Formatted I/O

Reading

You should read following topics before starting this exercise:

- Concept of streams in C++ (text and binary)
- C++ Stream Classes
- Concept of manipulators

Practical Assignment

Set A

1. Write a program which reads a text from the keyboard and displays the following information on the screen in two columns:
   - ✓ Number of lines
   - ✓ Number of words
   - ✓ Number of characters

   Strings should be left-justified and the numbers should be right-justified in a suitable field width.
2. Write a program to accept student details and print marksheet on screen using appropriate manipulators.

Assignment Evaluation

0: Not Done [ ] 1: Incomplete [ ] 2.Late Complete [ ]

3: Needs Improvement [ ] 4: Complete [ ] 5: Well Done [ ]

Signature of the Instructor **Date of Completion ____/____/______**

<u>Exercise 8</u> No. Of Slots: 3

Exception Handling

Reading

You should read following topics before starting this exercise:

- What is exception?
- C++ exception handling (try, catch, throw)
- What is the purpose of exception handling?
- std::exception

Practical Assignment

Set A

1. Write a C++ program that prompts the user for a positive no, If negative no. is entered, then throw an exception as no. not positive
2. Write a function count_letter that takes a single character. This function will count the number of consonants and vowels. If a non-letter is given to the function, it generates an exception

Set B

1. Write a C++ program for calculator which has operators as “+”,”-“,”*”,”/”. Display appropriate exceptions mesg. like divide by zero etc. and display the calculated results.
2. Write a C++ program that prompts the user to enter a sequence of numbers (floats) and then prints them in reverse order. Use stack. Give appropriate exception message as “Push not done - out of space."

Set C

1. Define appropriate exceptions for the Matrix class and its functions so that they throw exceptions when errors occur, including the following:
   - ✓ When the sizes of the operands of + and - are not identical.
   - ✓ When the number of the columns of the first operand of * does not match the number of rows of its second operand.
   - ✓ When the row or column specified for () is outside its range.
   - ✓ When heap storage is exhausted?

Assignment Evaluation

0: Not Done [ ]   1: Incomplete [ ]   2.Late Complete [ ]

3: Needs Improvement [ ]   4: Complete [ ]   5: Well Done [ ]

Signature of the Instructor   **Date of Completion** ____/____/______

<u>Exercise 9</u>

No. Of Slots: **3**

File I/O

Reading

You should read following topics before starting this exercise:

- Types of files supported in C++ : Text and Binary
- Difference between binary and text files
- File operations
- Concept of file pointers and their manipulations in C++
- File updations and random access

Practical Assignment

Set A

1. Write a program to count the number of characters, number of words and number of lines in a file.
2. Write a program which copies a user-specified file to another user-specified file. Your program should be able to copy text as well as binary files.
3. Write a user-defined function in C++ to read the content from a text file OUT.TXT, count and display the number of alphabets, words and lines present in it.

Set B

1. Assuming that a text file named FIRST.TXT contains some text written into it, write a function named copyupper(), that reads the file FIRST.TXT and creates a new file named SECOND.TXT contains all words from the file FIRST.TXT in uppercase.
2. Assuming that a text file named FIRST.TXT contains some text written into it, write a function named vowelwords(), that reads the file FIRST.TXT and creates a new file named SECOND.TXT, to contain only those words from the file FIRST.TXT which start with a lowercase vowel (i.e., with 'a', 'e', 'i', 'o', 'u').
   For example, if the file FIRST.TXT contains
   Carry umbrella and overcoat when it rains
   Then the file SECOND.TXT shall contain
   umbrella and overcoat it
3. Assuming the class Computer as follows :

```
class computer
{
        char chiptype[10];
        int speed;
        public:
                void getdetails()
                {
                        gets(chiptype);
                        cin>>speed;
                }
                void showdetails()
                {
                        cout<<“Chip”<<chiptype<<“ Speed= “<<speed;
```

```
            }
};
```

Write a function **readfile( )** to read all the records present in an already existing binary file SHIP.DAT and display them on the screen, also count the number of records present in the file.

4. Create a class CFile which has the following members
   fstream fp;
   char fn[MAXSIZE];
   Write necessary member functions and overload the following operators

Operators	**Purpose**	**Use**
+	F3=F1+F2	Puts contents of files F1 & F2 into F3
-	-F3	Changes the cases of all the upper and lower case letters

5. Given the following class declaration,

```
 class myFile
 {
      fstream in_out;
      public:
           myfiile(char * fname);
           void display()     // reads the files
           void append();  // append a character to a file
           void fileclose();  // close the file
 }
```

   Write a menu driven program with the following options
   - ✓ Display
   - ✓ Append
   - ✓ Exit

6. Write a C++ program to create a text file which stores employee. Objects(emp_id, emp_name, emp_sal). Write a menu driven program with the options
   - ✓ Append
   - ✓ Modify
   - ✓ Delete
   - ✓ Display
   - ✓ Exit

Set C

1. A file contains a list of telephone numbers in the following form:
   Ajay 12345
   Vijay 98765
   ------- --------
   The names contain only one word and the names and telephone numbers are separated by white spaces. Write a program to read the file and output the list in two columns. The names should be left-justified and the numbers right-justified.
2. Write an interactive, menu-driven program that will access the file created in the above program, and implement the following tasks.
   - ✓ Determine the telephone number of the specified person.
   - ✓ Determine the name if a telephone number is known.
   - ✓ Update the telephone number, whenever there is a change.

3. There are 100 records present in a file with each record containing a 6-character item code, a 20-character item name and an integer quantity. Write a program to read these records, arrange them in the ascending order and write them in the same file overwriting the earlier records.

Assignment Evaluation

0: Not Done [ ]   1: Incomplete [ ]   2.Late Complete [ ]

3: Needs Improvement [ ]   4: Complete [ ]   5: Well Done [ ]

Signature of the Instructor   **Date of Completion** ____/____/______

<u>Exercise 10</u> No. Of Slots: 3

Templates

Reading

You should read following topics before starting this exercise:

- Generic Programming in C++.
- Compile Time and Run Time Polymorphism and Templates.
- Concept of Templates in C++
- Function templates
- Class templates
- Overloading of function templates
- Function templates with multiple parameters
- Class templates with multiple parameters

Practical Assignment

Set A

1. Write a program in C++ to perform the following using the function template concepts:
   - ✓ To read a set of integers
   - ✓ To read a set of floating point numbers
   - ✓ To read a set of double members individually

   Find out average of the non-negative integers and also calculate the deviation of the numbers.
2. Implement a template function for quick sort.
3. Define a Swap function template for swapping two objects of the same type.
4. Define a class template stack which can be instantiated as follows:
   stack <int, 20> stk; //size of stack is 20.

Set B

1. Define template class for linked list with all necessary operations.
2. Using the Standard C++ Library **vector** as an underlying implementation, create a **Set** template class that accepts only one of each type of object that you put into it.

Set C

1. Using Standard Template Library(STL), write a program to obtain the list of files present in current directory and display these files in sorted order of name, type and size.

Assignment Evaluation

0: Not Done [ ]	1: Incomplete [ ]	2.Late Complete [ ]
3: Needs Improvement [ ]	4: Complete [ ]	5: Well Done [ ]

Signature of the Instructor **Date of Completion** ____/____/______

SECTION III
Database Case Study

CASE STUDY

Solve the case study according to the steps given below:

Step1: Choose the case study by selecting some real world problem.
Step2: Identify all the objects and their attributes from the selected case study.
Step3: Draw the class diagram.
Step4: Construct an SQL schema definition for the case study.
Step5: Construct the queries on the related case study.

Consider the following example:

Step 1: Consider a case study on desktop publishing system. In this schema, a document consists of numbered pages . Each page contains many drawing objects including ellipses, rectangles, polylines, and textlines. Ellipses and rectangles are embedded within a bounding box. A polyline is a sequence of line segments defined by vertex points. Textlines originate at a point and have a font. A group of objects is itself an object.

Step2: The following are the objects identified for the above case study.

- Document(page_width,page_height,left_margin,right_margin,pages*)
- Page(page-number,drawing-objects*)
- DrawingObject(line-thickness)
- Ellipse(bound-box)
- Rectangle(bound-box)
- BoundBox(left-edge,top-edge,width,height)
- Polyline(points*)
- Font(font-size,font-family,bold,italic,underline)
- GroupObjects(objects*)
- Point(x,y)
- Textline(point,font,text,alignment)

Step 3: Class diagram for desktop publishing system.

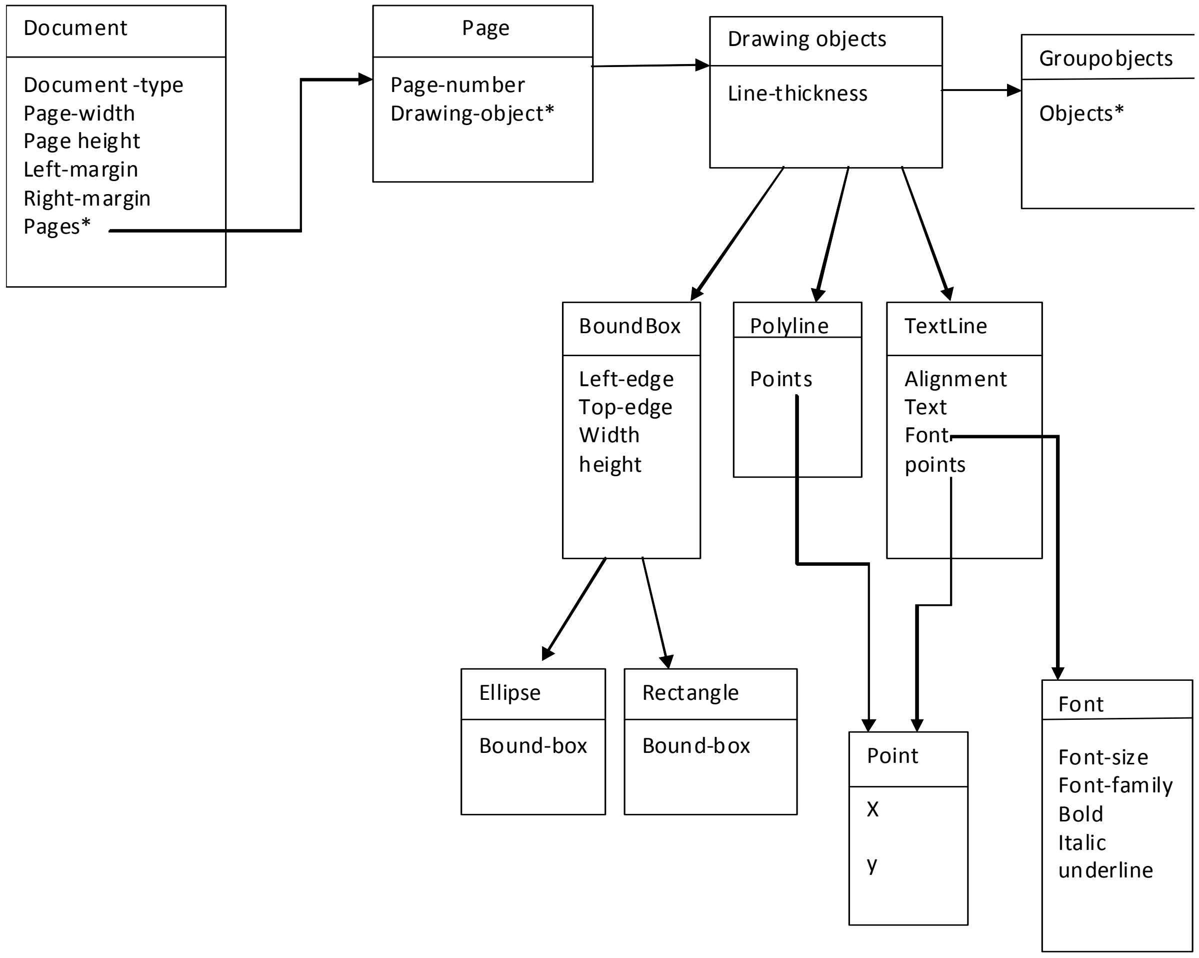


Fig: Desktop Publishing System

Step4: Construct a schema definition for the above case study as follows.

Construct the user defined data types:

Eg: 1. Create type pages as(page-number, drawing-object);

2. Create type document as(page-width, page-height, left-margin, right-margin, pages);

3. create type drawing object(line-thickness);

4. create type BandBox under drawing object(left-edge, top-edge, width, height);

Construct the table using user defined data types:

Eg: 1. create table bandbox of BandBox;
 2. create table ellipse of Ellipse under BandBox.

Step 5: Identify queries for the case study as follows:

The object-oriented SQL statements for the following queries:

a. List all document-types in the relation document.
b. List all the fonts of textline with size 2.
c. List all documents that have at least five pages.
d. Give all documents which have at least one page that has at least one bold text line.

Some solved examples:

Eg: a.
```
select distinct d.document-type
from document as d;
```

b.
```
select t.font,f.font-size
  from font as f, textline as t
  where font-size In( select font-size
                      from  font
                      where font-size=2);
```