I SEMESTER M.C.A.

OOP LAB MANUAL - 2023

(MCA – 4142)

Name : _____

Reg. No. : _____

MANIPAL INSTITUTE OF TECHNOLOGY
MANIPAL
(A constituent unit of MAHE, Manipal)

CONTENTS

Lab No.	Topic to be covered	Page No.	Date	Signature
	Course Objectives, Instructions to Students & Evaluation Plan			
1	Fundamentals of C++ programming and Logic building-I			
2	Logic building exercises-II and structures			
3	Functions, Function Overloading, and Arrays			
4	Introduction to OOP Concepts: Classes, Objects , constructors, destructors & Friend Functions			
5	Operator Overloading			
6	Operator Overloading and Data Conversion			
7	Inheritance			
8	Inheritance and Virtual Base class			
9	Virtual Functions and Polymorphism			
10	Files and Streams-I			
11	Files and Streams-II			
	Week 12 – End Term Practical Examination			
	Additional Exercises			

Reference Books:

1. Robert Lafore, "Object Oriented Programming in Turbo C++",Galgotia Publications Pvt.
2. K R Venugopal, Rajkumar, T Ravishankar, "Mastering C++",Tata McGraw Hill Publishing Company Ltd., 2005.
3. Dietel, H. M., & Dietel, P. J. (2010). C++ How to program. New York.

Course Objectives

1. To understand and learn to build programming logic
2. To develop programming skills using C++
3. To write, compile and debug C++ programs
4. To design and implement file handling programs

Instructions to Students

1. **Make entry in the Time Log Book as soon as you enter the Laboratory**.
2. All the students are supposed to enter the terminal number in the logbook.
3. Do not change the terminal on which you are working.
4. All students expected to get at least the algorithm of the program/concept to be implemented (before entering lab).
5. Observation book needs to be submitted regularly for evaluation.

Students *SHOULD NOT*:

1. Bring mobiles phones or any other electronic gadget to the lab
2. Leave the lab without prior permission from the faculty-in-charge.

Evaluation Plan

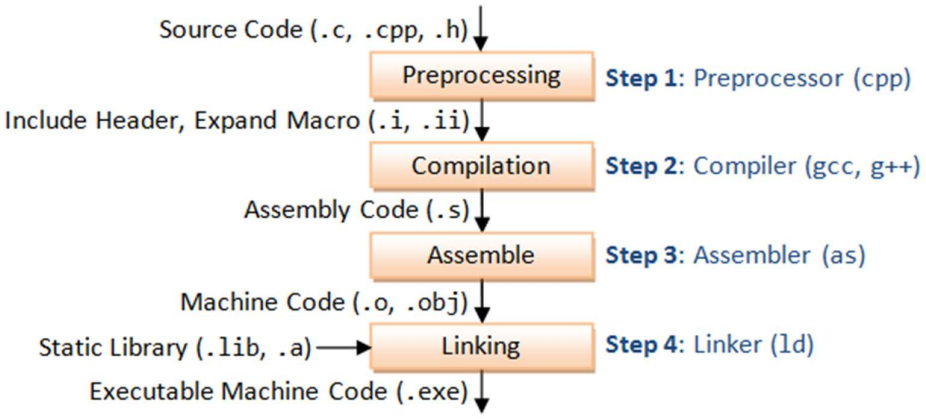
Continuous Internal Assessment (60 marks)	
3 evaluations	: 20 x 3 = 60 marks
Term End Assessment	: 40 marks
Total marks (IA+EA)	: 100 marks

Submitted by:

Dr. Poornima Panduranga Kundapur

Week 1: Fundamentals of C++ programming and Logic building-I

GCC Compilation Process:



Compiling and Executing a C++ Program (hello.cpp)

The C++ compiler that will be used in this lab is called **g++**. In order for a file to be recognized by the g++ compiler as a C++ program file, the name of the file name must have an appropriate extension. The extension to be used for program file names is **.cpp**. **You need to use g++ to compile C++ program.**

```
1. // hello.cpp
2. #include <iostream>
3. using namespace std;
4. int main() {
5. cout << "Hello, world!" << endl;
6. return 0;
7. }
```

I Simple compilation: [mca01@mca ~]$ **g++** ***progname*****.cpp**

If the program file contains no syntax errors, an executable file with the name **a.out** will be created. This file should not be displayed as it is not in a human readable form. The program can now be executed with the following command.
Command to run your program: [mca01@mca ~]$**./a.out**

If the program contained syntax errors, a series of error messages will be displayed. You must take note of the lines at which the errors occurred and then go back to the original program file (the .cpp file) to locate and correct the mistakes. Save your updated file. Then, recompile.

II Compilation with option -o:

We use the -o option to specify the output file name.

```
// Compile and link source  hello.cpp into executable hello.exe
[mca01@mca ~]$ g++ -o hello.exe hello.cpp
[mca01@mca ~]$hello      // Execute under CMD shell
[mca01@mca ~]$ ./hello   // Execute under Bash or Bourne shell,
                          //specifying the current path (./)
```

III Using "make" to compile:

An alternative method for compiling a C++ program is to use the Linux make utility. Using make to compile a program stored in the file **program.cpp**, will automatically invoke the g++ compiler using the –o option and create an executable file called **program**.

The command to compile and create the executable is:

[mca01@mca ~]$**make program**

Note that the .cpp extension should not be included in this command even though the name of the program file is program.cpp.

In order to execute the program, type the command:

[mca01@mca ~]$ **./program**

Some Good Programming Practices

- Every program **should begin with a comment** that explains the purpose of program, author and creation date and last updated date.
- Use blank lines, space characters and tabs to enhance program readability.
- Many programmers make the last character printed a newline (**`\n`**). This ensures that the function will leave the screen cursor positioned at the beginning of a new line.
- Indent the entire body of each function one level within the braces that delimit the body of the function. This makes the program easier to read.
- Place a space after each comma (**`,`**) to make programs more readable.
- Indenting the program and using braces for both blocks (if-else)

☞**Some Common Programming Mistakes**

- Forgetting to include the **`<iostream>`** header file in a program that inputs data from the keyboard or outputs data to the screen cause the compiler to issue an error message.
- Omitting the semicolon at the end of C++ statement is a syntax error.

☞**Writing C++ programs sing the Vi Editor:**

A C++ source code file is recognized by its extension like .cpp or .cc recommended by GNU.

Ideally, in C++ declarations are collected in header files with the extension .h. A C++ program normally consists of many classes that are defined in separate files. It must be

possible to compile the files separately. The program source code should be organized like
this (a main program that uses a class List):

- **Define the list class in a file list.h:**

```
#ifndef LIST_H // include guard
#define LIST_H
// include necessary headers here
class List {
public:
List();
int size() const;
...
private:
...
};
#endif
```

- Define the class member functions in a file list.cc:

```
#include "list.h"
// include other necessary headers
List::List() { ... }
int List::size() const { ... }
...
```

- Define the main function in a file ltest.cc:

```
#include "list.h"
#include <iostream>
int main() {
List list;
std::cout << "Size: " << list.size() << std::endl;
...
}
```

The “include” guard is necessary to prevent multiple definitions of names.

WEEK – 1 Exercises

1. Write a C++ program to exchange two variables without using any temporary variable.

2. Write a C++ program to find the largest of three numbers.

3. Write a C++ program to calculate the area of a circle and a triangle.
4. Write a C++ program to convert the time in seconds to hours, minutes and seconds.
5. Write a C++ program to convert the distance in mm to cm, inch, feet (1 cm = 10mm, 1 inch = 2.5cm, 1 feet = 12 inches).
6. Write a C++ program to convert the temperature given in Fahrenheit to Celsius and vice versa. (C = 5/9(F-32))
7. Write a C++ program to calculate the compound interest.
   $A = P (1 + r/n)^{nt}$ *[Hint after bonus question]*
   Where, A = the future value of the investment/loan, including interest
   P = the principal investment amount (the initial deposit or loan amount)
   r = the annual interest rate
   n = the number of times that interest is compounded per unit t
   t = the time the money is invested or borrowed for
8. Write a C++ program to accept student details such as Name, Registration number, Year of Joining, Semester number, marks in five subjects. Calculate the average marks as total marks divided by five. Design a score card as based on the following grading criteria:

Average > =90	Grade A
Average between 80 and 89	Grade B
Average between 61 and 79	Grade C
Average between 51 and 59	Grade D
Average between 41 and 49	Grade E
Less than 40	Grade F (Fails)

Score Card for Student: John Smith

Registration Number: 1500009199 Semester: I Year: 2016
Grade Assigned:

Serial No.	Subject Name	Marks Scored (out of 100)
1	Object Oriented Programming	87
2	DBMS	90
3	Research Methodology	89
4	Computational Mathematics	80

5	Web Technologies		88
		Total:	434
		Average:	86.5
		Grade:	B

Bonus Question:

1. Write a C++ program, which generates all the possible combinations of the given 3-digit number.

Hint: *While compiling, ensure that you use the option –ln with the general compilation command.*

Pre-lab Exercises: Week 2

Analyze the following for loops in the table and state whether (a) they compile (b) they run (c) They print out a line (d) they print out more than one line. **Answer yes or no.**

No.	Action	Compile?	Run?	Print out a line?	Prints out multiple lines?
1	for (i = 1; i <=10; i++); cout<<i<<endl;				
2	for (j = 11; j < 11; ++j); cout<<j<<endl;				

3	int j=100;
	for (int j = 1; j <=10;
	j++);
	cout<<j<<endl;				

4	for (j = 1; j <= 10; j++) cout<<"Hello\n";				

5	for (j = 1; j <= 10; j++)
	{
	if(j < 5) continue;
	cout<<j<<"\n";
	}				

6	for (j = 1, j <= 10, j++) cout<<"Hello\n";				

Week 2: Logic building exercises-II and structures

1. Write a C++ program to print the following patterns (number of lines as user input): (Use nested for loops & also try with nested while loops)

* * * * * * * * * *	A B C D A B C A B A	1 2 3 4 5 6 (Floyd's triangle)

2. Write a C++ program to check whether the given number is a perfect cube or not (without using any library function). Display the proper message accordingly. (Example: 27 is a perfect cube, cube root is 3.)
3. Write a C++ program to convert a decimal number to its equivalent binary format. (Example: 25 would be 11001) and vice versa.
4. Write a C++ program to generate the first ***n*** terms of the Fibonacci sequence.
5. Write a C++ program to generate prime numbers between two limits.
6. Write a C++ program to implement the solution to *change making problem*.
   Change making problem: finding the minimum number of coins (of certain denominations) that add up to a given amount of money.

Example: Suppose the total bill = ₹ 90, cash paid = ₹ 500. So, balance = 500 - 90= ₹ 410
The optimal solution: Balance amount to be generated: { ₹ 200, ₹ 200, ₹10 }.

7. Define a structure to represent the bank account of a customer, where the members are *Customer name, Account number, Account type (Savings, Fixed or Current) and balance amount*. Perform the following operations:
    i. Deposit an amount.
    ii. Withdraw an amount after checking the balance.

☞**BONUS QUESTION:**

Write a C++ program to display first ***n*** decimal numbers whose binary equivalent contain even number of one's.

Pre-lab Exercises: Week 3

Analyze the following for concepts and state whether (a) they compile (b) they run (c) An error occurs(d) they print output. **Answer yes or no.**

No.	Action	Compile?	Run?	Error?	Output?
1	float average(x,y);				

2	void f()
	void f(){ cout<<"C++"; }
	int main(){
	f();
	return 0;
	}				

3	void f(int, int);
	void f(int x, y){
	x =1; y=2;}
	int main(){
	int x=10;
	f(x,x);
	cout<< x;
	return 0;
	}				

4	int i; double *dptr; dptr= &i;				
5	int array[] = {10, 20, 30}; cout << -2[array];				

6	void display(int a[][], int m, int n)
	{
	int i,j;
	for (i=0; i<m; i++)
	for (j=0; j<n; j++)
	cout<<""<<a[i][j];
	cout<<"\n";
	}				

Week 3: Functions, Function Overloading and Arrays

1. Write a program to compute cube of a number using inline function.
2. Write a function to determine if the given string is a palindrome or not. Accept string in the main function and pass that as a parameter to the function (use c-style strings)
3. Write an interactive program in C++ for swapping two integers, floats, and characters using function-overloading concept.

4. Write a C++ program that computes the inverse of an integer and the double of the inverse. Example: Inverse (1367) = 7631 and double (7631) = 15262. Use functions for calculating inverse and double of inverse.

5. Given that an EMPLOYEE structure contains following members. Data members: Employee_Number, Employee_Name, Basic, DA, IT, Net_Sal. Write functions: To read the data, calculate net salary and display the net salary. Write a C++ program to read the data of N employees and compute Net salary of each employee. (DA= 12% of Basic and Income Tax (IT) = 18% of the gross salary).

☞ **BONUS QUESTION:**

1. Write a C++ program to calculate *mean, standard deviation, variance* of the input data points.
   Example:

i	1	2	3	4	5	6
$\mathbf{X_i}$	3	4	4	5	6	8

$$\mu = mean(X) = \frac{1}{N}\sum_{i=1}^{N} X_i$$

$$variance(X) = \sigma^2 = \frac{1}{N}\sum_{i=1}^{N} (X_i - \mu)^2$$

Mean(X) = 5, Std_deviation(X) = 2.67, var(X) = 1.63

2. Solve using arrays only (may be more than one dimension)

   A hotel consists of 5 floors. Each floor has 8 rooms numbered 1 to 8. Accept the number of people staying in each room on each floor. Display the following information:
   Total Number of occupants =
   Total Number of adults =
   Total Number of children =
   Total Floor wise occupancy =
   Total Floor wise adult occupancy=
   Total Floor wise child occupancy=
   Max. Occupancy is on _____ floor
   Min. Occupancy is on _____ floor

3. Given a list of 2D-points: (x_1,y_1), $(x_2,y_2)\ldots(x_n,y_n)$. Find the point in the list that is nearest to the input point (x_k,y_k) based on the minimum distance.
   Note: The distance between two points $P(x_1, y_1)$ and $Q(x_2, y_2)$ is calculated as:

   $$d(P,Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Week 4: Classes, Objects & Friend Functions

1. Create a **flight** class that has private data members: flight number (integer), destination (characters), distance (float), fuel (float).
   a) Provide a default constructor that initializes distance to 500 kms always
   b) ***Private Member*** functions: calculate_fuel()to calculate the value of Fuel as per the following criteria:

Distance (in kilometers)	Fuel (in liters)
<=1000	500
>1000 and <=2000	1100
>2000	2200

   c) ***Public Member*** functions: information_entry() to allow user to enter values for flight number, destination, distance which calls function calculate_fuel() to calculate the quantity of fuel and display_info() to allow viewing flight details.
2. Define a class to represent the **bank account** of a customer where the data members are Customer name, Account number, Account type [Savings(S), Fixed(F) or Current(C)] and balance amount.
   Member functions:
   i. To assign initial opening balance to Rs. 500 using constructor
   ii. To deposit an amount into the account
   iii. To withdraw an amount after checking the balance. *(Use friend function)*
   iv. To display the account details.
   Write the main function to test the above program with two objects.
3. Write necessary class and member function definitions for a cricket player object. The program should accept the details from user (maximum 10 objects). The details of the player are player code, name, number of matches played, total runs scored in all matches 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 total runs
   (Use function overloading and appropriate constructors)
   Sample input for a player (similar for the remaining players of the team)
   Player code: IND001
   Player name: Rahul Dravid
   Number of matches played: 150
   Total runs: 7500
   Number of times not out: 100
4. Write a C++ program to create a class called **Complex** and implement the following by overloading functions that return the complex number.
   i. add (s1,s2) where s1 is an integer and s2 is a complex number
   ii. add (s1,s2) where s1 & s2 are both complex numbers.

Week 5: Operator Overloading

1. Write a C++ program to create a class called **Matrix** using 2-Dimensional array of integers. Implement the following by overloading the operator == which checks the compatibility of the two matrices to be added and subtracted. Perform the following by overloading + and – operators.

```
if ( m1 == m2 )
{
        m3 = m1 + m2;
        m4 = m1 – m2;
}
else    {               // Display an error message

                        } where m1, m2, m3, and m4 are objects of Matrix class.
```

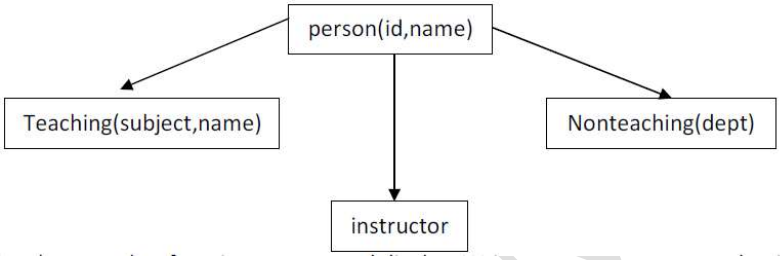
2. Write a C++ program to create a class called **Date**, which contains integer members to represent day, month, and year. The overload ++ operator to increment the value of Date object by one.

Week 6: Operator Overloading and Data Conversion

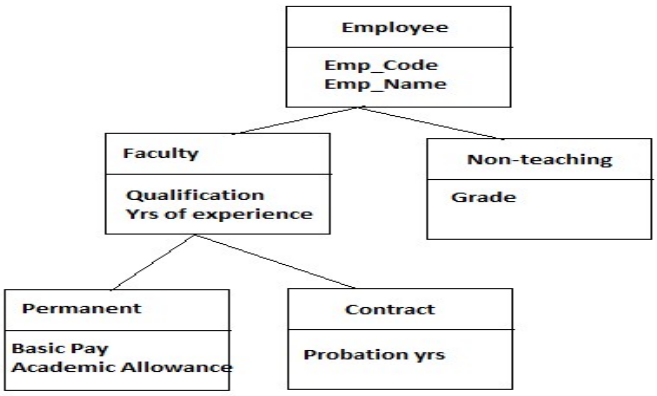
1. Write a C++ program to overload the relational operators <= to compare 2 objects of **my_String** class
2. Write a C++ program to perform addition and subtraction on two complex numbers using operator overloading using member functions.

Week 7: Inheritance

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.

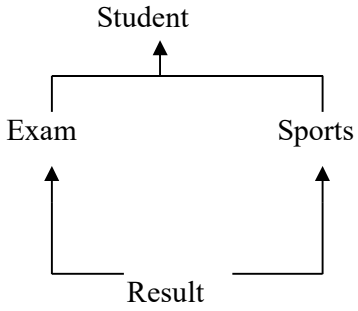


2. Consider a college that maintains records of its employees. The hierarchical relationship of this employee database is divided into a number of classes as shown in the figure given below. It contains the data needed for each class. Define functions to create the database and retrieve information as needed.



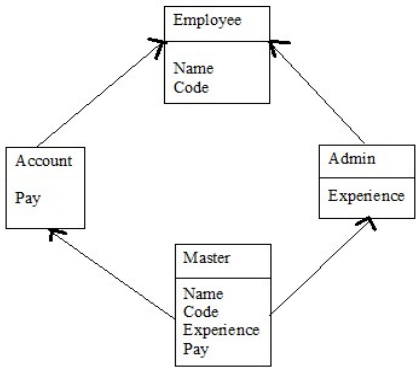
Week 8: Inheritance and Virtual Base class

1. A new scheme of evaluation of student's performance is formulated that gives weightage for sports. The relationships of different classes and derived classes are given below.



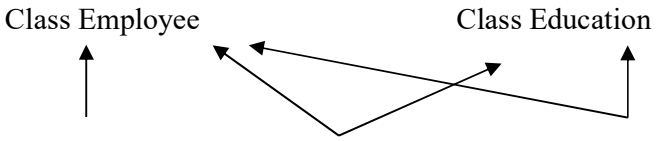
By properly assuming the data members & member functions for each class, write a C++ program to model the above relationship such that members of the student class are not inherited twice.

2. Consider the following: The class Master derives information from both Account and Admin classes, which in turn derive information from the class Employee. Define all the 4 classes and write a program to create, update, and display the information contained in the Master object.



Week 9: Virtual Functions and Polymorphism

1. Consider the following class hierarchy:



   Write an interactive C++ program to model the above relationship. Assume proper data members and member functions for each class.

2. Create a base class called **shape**. Use this class to store two double type values that could be used to compute the area of figures. Derive two classes called **triangle** and **rectangle** from shape. Add to the base class, a member function **get_data()** to initialize base class data members and another member function **display_area()** to compute and display the area of figures. Make **display_area()** as a virtual function and re-define this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle of a rectangle interactively, and display the area. (Hint: The two input values can be treated ad lengths of two sides for rectangle and as base and height for triangle)

Week 10: Files and Streams -I

1. Write a menu driven C++ program to create a single line text file and then implement the following:
   a) Convert every lower-case letter to upper case and upper-case letter to lower case.
   b) Display the content of the file.
   c) Determine how many times a character ‘a’ is present in the file.

2. Write a C++ program to create a multiline text file and then determine the following: Total number of characters, letters of the alphabet, digits, lines and spaces.

Week 11: Files and Streams -II

1. Write a C++ program to create an employee database and store the following information: Name, Employee id, Department, Designation, Basic salary. Then accept an employee id from the keyboard and generate the salary slip for that employee. Use suitable formulas to calculate DA, HRA, GROSS PAY, INCOME TAX and NET PAY. Make use of random-access file concept. (Employee id should be generated automatically).

2. Write a C++ program to create an item database and store the following information:
   i. Item name ii. Item code iii. Unit price iv. Quantity on hand
   Then implement the following:
   i. Add a new record
   ii. Modify the existing record
   iii. Delete a record.
   Use random access file concept. The item code should be generated automatically.

Additional Exercises

1. Write a C++ program to generate the Pascal's triangle:

```
          1
        1   1
      1   2   1
    1   3   3   1
```

2. Write a C++ program to generate all the prime numbers between 1 and n , where n is a value supplied by the user.
3. Write a C++ program to read a string input, a key k, 1 <= k <= 25 and then encrypt that. Verify your program, with the input "Rome is not built in one day", the output will be "Urph lv qrw exlow lq rqh gdb" if key is 3.
4. Write a C++ program to decipher a string encrypted by the encryption program in above exercise. Verify program with above output.
5. Write a C++ program to calculate the electricity bill for a consumer. The rates for domestic user consumption is given below:
   For first 100 units: 0.60 paise per unit
   For next 200 units: 0.80 paise per unit
   Above 300 units : 0.90 paise per unit
   All users are charged minimum of Rs. 50. If total amount is more than Rs. 300 then an additional surcharge of 15% is added. Accept customer name and number of units consumed and print out the total bill.
6. Write a C++ program to create a class, which keeps track of the number of its instances. Use static member, constructors and destructors to maintain updated information about active objects.
7. A point on the two-dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. The sum of two points can be defined as a new point whose X coordinate is the sum of the X coordinates two points and same with the Y coordinate. Write a program using a C++ structure **point** to model with two data members X and Y coordinates and to find the sum of two points in a third point. Use a member function to accept values and calculate the sum of 2 points and display the X and Y coordinates of the 3 points.
8. Create a class, which contain a vector (a series of float values) as its data member. Include member function to perform the following:
   i. To create the vector.
   ii. To modify the value of a particular element.
   iii. To multiply the vector by a scalar value.
   iv. To display the vector in a neat format.
   v. Addition of 2 vectors & display the resultant vectors.
   Write a C++ program to test your class.
9. Modify Exercise E that defines a structure **point** to now include a class named **point** with two data members X and Y coordinates. Include a constructor to initialize the member data and as well as a parameterized constructor. Member functions include

accept and display coordinate values. Additionally add a member function that calculates the sum of two points in a third **point** object.

10. Create a class **circle** with data members x, y and r where x and y are coordinate points at the center and r is the radius. Include member functions to find area and circumference of the circle. Initialize the object of the class using parameterized constructor. Write a C++ program for the above class.
11. Define a non-member function circ_area() that calculates the area of a circle using the above defined class **circle**.
12. Write a C++ program to compute the Frobenius norm of a matrix.

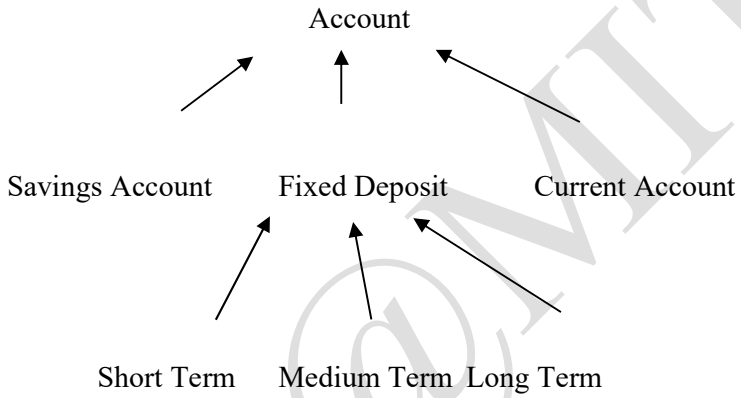
    Example: $A = \begin{bmatrix} 1 & 2 \\ 5 & 7 \end{bmatrix}$

    Frobenius norm of A = $\sqrt{1^2 + 2^2 + 5^2 + 7^2}$ = 8.8882
13. Create a class called Array, which contains 1-Dimensional array of integers and an integer member which represents the actual size of the array. The Array class contains the following member functions:
    i. void inputarray ()
    ii. void display ()
    iii. int largest ()
    iv. float average ()
    v. void sort ()

    Write a C++ program to implement and test the above class.
14. Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a fifty-paise toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have passed by and the total amount of money collected. Model this tollbooth with a class called tollbooth. The two data members are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to zero. A member function called payingCar() increments car total and adds 0.50 to the cash total. Another function called nopayCar() increments the car total but adds nothing to the cash total. Finally a member function called display() displays the two totals. Write a program to test this class. The program should allow user to push one key to count a paying car and another to print out the total cars and total cash and then exit.
15. Write a C++ program to perform addition and subtraction on two complex numbers using operator overloading using friend functions.
16. Write an interactive operator overloaded program in C++ for manipulating objects of Matrix class. Overload operators such as >>, <<, +, *, ==.
17. Write a C++ program to convert **Distance** (object) to meters (float) and vice-versa using type casting technique. Overload proper type casting operator.
18. Write a C++ program to convert Rupees (object) to paise (integer) and vice-versa using type casting technique. Overload proper type casting operator.
19. Imagine a publishing company that markets both books and DVDs whose activities are shown in the above fig. Create a class **Publication** that stores the title (string) and price (float) of a publication. From this class derive two classes **Book** which

adds page count (type int) and **DVD** which adds storage (type int). Each of these three classes should have a getdata() function to get its data from the user and putdata() function to display the data. Write a C++ program to test the classes **Book** and **DVD** by creating their instances.

20. Modify above exercise 16 as follows: Add a base class **Sales** that holds an array of 3 floats so that it can record the sales of a particular publication for the last 3 months. Include a getdata() function to get three sales amounts from the user and a putdata() function to display the sales figures. Alter the **Book** and **DVD** classes so they are derived from both **Publication** and **Sales**. An object of class **Book** or **DVD** should input and output sales data along with its other data. Write a main() function to create a **Book** object and a **DVD** object and exercise their input-output capabilities.
21. Consider the following relationship:



Write an interactive C++ program to model the above relationship. Use appropriate data members & member functions for each class.

22. Create a class called **time** that has separate **int** member data for hours, minutes and seconds. One constructor should initialize these data to 0, and another should initialize it to fixed values. A member function should display it in **hh:mm:ss** format. The final member function should add two objects of Time passed as arguments. A main () program should create 2 initialized **time** objects and one that is not initialized. Then it should add the 2 initialized objects leaving the result in the third **time** object. Finally it should display the values of all the objects.
23. Write a menu driven C++ program to create a multiline text file. Then implement the following:
    a) Copy the content of the file to another file by suppressing all spaces.
    b) Perform encryption and decryption.
    c) Display the contents of the file by suppressing all special characters.
24. Write a menu driven C++ program to create a multiline text file (using string I/O), and then implement the following:
    a) Display the file contents by converting the first character of each word to capital letter.
    b) Determine the total number of words and total number of spaces present in the file. Note that the words can be separated by more than one space.
    c) Determine the total number of palindromes present in the file.
25. Write a C++ program to create a book database and store the following information:

i. Title ii. Author name iii. Accession number iv. Pages

Then display the book information based on the following:

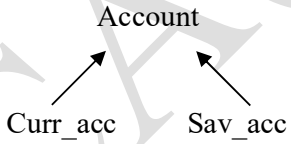
i. Title ii. Author iii. Accession number.

Make use of sequential file concepts. (Accession number must be auto generated)

26. Write a C++ program to read a sequence of strings from the standard input until either the same word occurs twice in succession or all the words have been read. Use a while loop to read the text one word at a time. Use the **break** statement to terminate the loop if a word occurs twice in succession. Print the word if it occurs twice in succession, or else print a message saying that no word is repeated. (Apply user-defined functions wherever applicable)
27. Write a C++ program to create a student database and store the following information:

    i) Name ii) Roll number (auto generated) iii) Branch iv) Marks in 6 subjects.

    Then accept a student roll number from the keyboard and generate the marks sheet for that student. Make use of sequential file concept.
28. Write an interactive C++ program to create a graphic class hierarchy. Create a base class called Figure and derive two classes Close and Open from that. Define two more classes called Polygon and Ellipse using the Close class. Create derived classes Line and Polyline from the Open class. Define three objects (triangle, rectangle, and pentagon) of the class Polygon. All classes must have appropriate member function including constructors and destructors.
29. Create class called **Float** that contains one float member data. Overload all the 4 arithmetic operators so that they operate on objects of Float class.
30. Consider the following relationship:



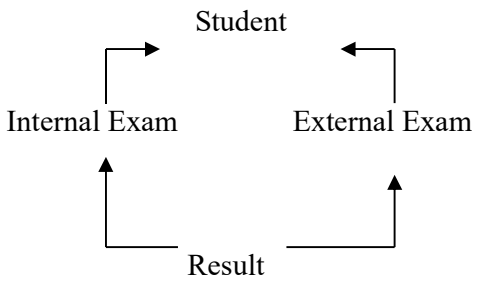
Account class has name, account number, and account type as its data members, getdata(), display() as its member functions.

Curr_acc class has balance and minimum as its data members, getdata(), deposit(), withdrawl(), interest(), displaybalance() as its member functions.

Sav_acc class has balance and minimum as its data members, getdata(), deposit(), withdrawl(), interest(), displaybalance() as its member fumctions.

Write an interactive program to model the above relationship.

31. Consider the following relationship:



Write an interactive C++ program to model the above relationship. Assume proper data members and member functions for each class. Include data validation. Also display the grade-sheet. Test your program for 10 students.

Faculty Remarks

Evaluation Sheet

Evaluation	**E1** (20)	**E2** (20)	**E3** (20)	**Out of 60**
Marks				
Date				
Remarks				
Signature				