Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2021-22

(310902) Data Structure and Algorithm Laboratory

****

Class: FY-MCA Shift / Div :B Roll Number :56

Name:Ayush Kishor Raut Assignment No : 2 Date of Implementation :19/12/2022

****

Q.1. 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

Program:

```
#include<iostream>

using namespace std;

class Account

{

   char accname[30];

   int accno;

   char type[10];

   int balance;

public:

   void init();

   void deposit();

   void withdraw();

   void display();

};

void Account::init()
```

```
{
   cout<<"\nEnter the Account holder's Name: "<<endl;
   cin>>accname;
   cout<<"\nEnter the Account Number: "<<endl;
   cin>>accno;
   cout<<"\nEnter the Type of Account: "<<endl;
   cin>>type;
   cout<<"\nEnter the Balance in Account: "<<endl;
   cin>>balance;
}
void Account::deposit()
{
int amount = 0;
cout<<"\n\t\tYou are here to deposit the amount\n"<<endl; cout<<"\nEnter the amount to deposit in the Account: "<<endl;
cin>>amount;
balance += amount;
}
void Account::withdraw()
{
   int amount = 0;
   cout<<"\n\t\tYou are here to withdraw the amount\n"<<endl;
   cout<<"\nEnter the amount to withdraw from the Account: "<<endl;
   cin>>amount;
   if(amount >= balance)
      cout<<"\tInsufficient Balance"<<endl;
   else
      balance -= amount;
}
```

```
void Account::display()
{
cout<<"Account Number: "<<accno<<endl;
cout<<"Account Holder's Name: "<<accname<<endl;
cout<<"Type of Account: "<<type<<endl;
cout<<"Balance: "<<balance<<endl;
}
int main()
{
Account A;
cout<<"\n------------------Welcome to Modern Bank------------"<<endl;
   A.init();
   A.deposit();
   A.withdraw();
   A.display();
   return 0;
}
```

Output:

```
------------------Welcome to Modern Bank------------

Enter the Account holder's Name:
Ayush
Enter the Account Number:
500
Enter the Type of Account:
Saving
Enter the Balance in Account:
50000
                  You are here to deposit the amount
```

```
Enter the amount to deposit in the Account:
500
                    You are here to withdraw the amount
Enter the amount to withdraw from the Account:
500
Account Number: 500
Account Holder's Name: Ayush
Type of Account: Saving
Balance: 50000
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2021-22

(310902) Data Structure and Algorithm Laboratory

****

Class: FY-MCA Shift / Div :B Roll Number : 56

Name:Ayush Kishor Raut Assignment No : 2 Date of Implementation : 19/12/2022

****

Q.2. Write a C++ program to make an arithmetic calculator using an inline function.

Program:

```
#include<iostream>
using namespace std;
int Add(int a, int b)
{
return a + b;

}
int Sub(int a, int b)
{
return a-b;
}
int Multiply(int a, int b)
{
return a*b;
}
int Div(int a, int b)
{
return a/b;
```

```
}
int Modulo(int a, int b)
{
return a%b;

}
int main()
{
int num1 = 0, num2 = 0; int choice = 0;
cout<<"\n----------------Welcome to Sun Enterprises---------"<<endl;
    cout<<"\nEnter first number: "<<endl;
cin>>num1;
cout<<"\nEnter second number: "<<endl;
cin>>num2;
do {
cout<<"\n\t--AvailableOperations---\n\n1.Addition\n2.Subtraction\n3.Multiplication\n4.
Division \n5.Modulous\n6.Exit"<<endl;
cout<<"What would you like to perform?: ";
    cin>>choice;
cout<<endl;
switch(choice)
{

 case 1:
cout<<"\t\tAddition\t"<<endl;
cout<<"Sum is: "<<Add(num1,num2)<<endl;
break; case 2:
cout<<"\t\tSubtraction\t"<<endl; cout<<"Subtraction is: "<<Sub(num1,num2)<<endl;
        break;
case 3:
```

```
cout<<"\t\tMultiplication\t"<<endl;
cout<<"Multiplication is: "<<Multiply(num1,num2)<<endl;
        break;
case 4:
cout<<"\t\tDivision\t"<<endl;
cout<<"Division is: "<<Div(num1,num2)<<endl;
break;
    case 5:
cout<<"\t\tModulous\t"<<endl;
cout<<"Remainder is: "<<Modulo(num1,num2)<<endl;
        break;
case 6:
cout<<"\t\tExit\t"<<endl;
cout<<"You have exited from application"<<endl;
break;
    default:


cout<<"\nInvalid choice"<<endl;
}
}
    while(choice > 0 && choice < 6);
}
```

Output:

```
----------------Welcome to Sun Enterprises---------
Enter first number:
25
Enter second number:
10
            --AvailableOperations---
1.Addition
```

```
2.Subtraction
3.Multiplication
4. Division
5.Modulous
6.Exit
What would you like to perform?: 1
                    Addition
Sum is: 35
          --AvailableOperations---
1.Addition
2.Subtraction
3.Multiplication
4. Division
5.Modulous
6.Exit
What would you like to perform?: 2
                    Subtraction
Subtraction is: 15
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2021-22

(310902) Data Structure and Algorithm Laboratory

****

Class: FY-MCA Shift / Div :B Roll Number : 56

Name:Ayush Kishor Raut Assignment No : 2 Date of Implementation : 19/12/2022

****

Q.3. Write a program to find largest among given 3 numbers using inline function.

Program:

```
#include<iostream>
using namespace std;
void findLargenum(int a, int b, int c)
{


   if(a > b && a> c)
      cout<<"A is largest"<<endl;
   else if(b > a && b > c)
      cout<<"B is largest"<<endl;
   else
      cout<<"C is largest"<<endl;
}
int main()
{
int num1 = 0, num2 = 0, num3 = 0;
cout<<"\nEnter first number: ";
cin>>num1;
cout<<"\nEnter second number: ";
cin>>num2;
cout<<"\nEnter third number: ";
cin>>num3;
findLargenum(num1, num2, num3);
    return 0;
}
```

Output:

```
Enter first number: 45
Enter second number: 75
Enter third number: 95
C is largest.
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2021-22

(310902) Data Structure and Algorithm Laboratory

****

Class: FY-MCA Shift / Div :B Roll Number : 56

Name:Ayush Kishor Raut Assignment No : 2 Date of Implementation : 19/12/2022

****

Q.4. Write an inline function to obtain the largest of three numbers.

Program:

```
#include<iostream>
using namespace std;
int findLargenum(int a, int b, int c)
{
if(a > b && a > c) return 1;
else if(b > a && b > c) return 2;
else
return 3;
}
int main()
{
int num1 = 0, num2 = 0, num3 = 0;
cout<<"\nEnter first number:";
cin>>num1;
cout<<"\nEnter second number: ";
cin>>num2;
cout<<"\nEnter third number: ";
cin>>num3;
int value = findLargenum(num1, num2, num3);
if(value == 1)
cout<<"A is largest"<<endl;
else if(value == 2)
cout<<"B is largest"<<endl;
else
cout<<"C is largest"<<endl;
return 0;
}
```

Output:

```
Enter first number:85
Enter second number: 45
Enter third number: 25
A is largest.
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2021-22

(310902) Data Structure and Algorithm Laboratory

****

Class: FY-MCA Shift / Div :B Roll Number : 56

Name:Ayush Kishor Raut Assignment No : 2 Date of Implementation : 19/12/2022

****

Q.5. Create a class distance which accept the distances in feet & inches from the user & display the sum of two distances in feet & inch. [use object as function argument, Write the function inside the class] Modify above program to return the distance as an object of the same class.

Program:

```
#include<iostream>
using namespace std;
class Distance
{
float feet; int inch; public:
void init();
Distance sum(Distance, Distance); void show(Distance);
};
void Distance::init() {
cout<<"\nEnter distance in feet: "; cin>> feet;
cout<<"\nEnter distance in inch: "; cin>>inch;
 }
Distance Distance::sum(Distance si, Distance ki) {
Distance ri;
ri.feet = si.feet + ki.feet; ri.inch = si.inch + ki.inch;
return ri; }
void Distance::show(Distance ri) {
cout<<"\n\nTotal distance in feet: "<<ri.feet<<"\nTotal distance in inch: "<<ri.inch<<endl;
}
```

```
int main() {
Distance si, ki, io;
si.init();
ki.init();
io= si.sum(si,ki);

si.show(io);
    return 0;

}
```

Output:

```
Enter distance in feet: 35
Enter distance in inch: 22
Enter distance in feet: 55
Enter distance in inch: 26
Total distance in feet: 90
Total distance in inch: 48
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2021-22

(310902) Data Structure and Algorithm Laboratory

****

Class: FY-MCA Shift / Div :B Roll Number : 56

Name:Ayush Kishor Raut Assignment No : 2 Date of Implementation : 19/12/2022

****

Q.6. Write a C++ program to concatenate two strings by showing the use of dynamic constructor.

Program:

```
#include <iostream>
#include <cstring>
using namespace std;
class String
{
private:
   char *string_array;
public:
   String(const char *in_str)
   {
      int SIZE_STRING = sizeof(in_str);
      string_array = new char[SIZE_STRING + 1];
      strcpy(string_array,in_str);
   }
   void concat(String sobj)
   {
      int l1, l2;
      l1 = strlen(string_array);
```

```
        l2 = strlen(sobj.string_array);
        char *str_temp;
        str_temp = new char[l1 + 1];
        strcpy(str_temp, string_array);
        delete [] string_array;
        string_array = new char[l1 + l2 + 1];
        strcpy(string_array, str_temp);
        strcat(string_array, sobj.string_array);

    }
    void display()
    { cout<<string_array<<endl;
    }
    ~String()
    {
        delete [] string_array;

    }
};
int main()
{
String s1("hello "), s2(" world!");
s1.display();
s2.display();
s1.concat(s2);
s1.display();
return 0;
}
```

Output:

```
hello
```

```
world!
hello  world!
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2021-22

(310902) Data Structure and Algorithm Laboratory

****

Class: FY-MCA Shift / Div :B Roll Number : 56

Name:Ayush Kishor Raut Assignment No : 2 Date of Implementation : 19/12/2022

****

Q.7. Write a C++ program to solve Quadratic equations using constructor.

Program:

```
#include <iostream>
#include <cmath>
using namespace std;
class Roots{
int a, b, c; float r1, r2;
public:
void getData();
int determinant();
void checkdeterminant(int);
};
void Roots::getData() {
cout<<"\nEnter coefficient of x^2: "; cin>>a;
cout<<"Enter coefficient of x: ";
 cin>>b;
cout<<"Enter coefficient of 1: "; cin>>c;
}
int Roots::determinant() {
int d = b * b;
```

```
d -= (4 * a * c); return d;
}
void Roots::checkdeterminant(int d) {
if(d == 0) {
cout<<"Real and equal roots\n: "; r1 = (-1 * b);
r1 /= (2 * a);
r2 = r1;
cout<<"Roots: "<<r1<<" and "<<r2<<endl; }
else if(d > 0)

 {
cout<<"Real and distinct roots\n";
r1 = (-1 * b) + sqrt(d);
r1 /= (2 * a);
r2 = (-1 * b) - sqrt(d);
r2 /= (2 * a);
cout<<"Roots: "<<r1 <<" and "<<r2<<endl;
}
else{
cout<<"Imaginary roots"<<endl;
r1 = (-b) / (2 * a);
r2 = (sqrt(-d)) / (2 * a); cout<<"Roots: "<<r1 <<" + i"<<r2; cout<<" and "<<r1<<" - i"<<r2<<endl;
} }
int main(){
Roots r;
r.getData();

cout<<endl;
int d {0};
```

```
d= r.determinant(); r.checkdeterminant(d);
cout<<endl;
return 0;

}
```

Output:

```
Enter coefficient of x^2: 1
Enter coefficient of x: 4
Enter coefficient of 1: 4
Real and equal roots
: Roots: -2 and -2
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310907) OOP Laboratory

****

Class: FY-MCA Shift / Div : A Batch: f1 Roll Number : 51018

Name: Rudrani D Gaikwad Assignment No : 6 Date of Implementation : 27/02/23

****

Q1.Write a program to implement I/O operations on characters. I/O operations includes inputting a string, Calculating length of the string, Storing the string in a file, fetching the stored characters from it, etc.

Program:

```
#include <fstream>
#include <cstring>
#include<iostream>
using namespace std;

int main() {
    char input_string[100];


    cout << "Enter a string: ";
    cin.getline(input_string, 100);

    int string_length = strlen(input_string);
    cout << "Length of the string is: " << string_length << endl;


    ofstream outfile;
    outfile.open("output.txt");
    outfile << input_string;
    outfile.close();


    char output_string[100];
    ifstream infile;
    infile.open("output.txt");
    infile.getline(output_string, 100);
    infile.close();


    cout << "String fetched from file: " << output_string << endl;

    return 0;
}
```

Output:

```
Enter a string: hello
Length of the string is: 5
String fetched from file: hello
```

Q2.Write a program to copy the contents of one file to another.

Program:

```
#include <iostream>
#include <fstream>
```

```
using namespace std;

int main() {
    string source_file, destination_file;
    ifstream source_stream;
    ofstream destination_stream;


    cout << "Enter the name of the source file: ";
    getline(cin, source_file);


    source_stream.open(source_file.c_str());
    if (!source_stream.is_open()) {
        cerr << "Error opening source file." << endl;
        return 1;
    }

    cout << "Enter the name of the destination file: ";
    getline(cin, destination_file);


    destination_stream.open(destination_file.c_str());
    if (!destination_stream.is_open()) {
        cerr << "Error opening destination file." << endl;
        return 1;
    }

     char c;
    while (source_stream.get(c)) {
        destination_stream.put(c);
    }


    source_stream.close();
    destination_stream.close();

    cout << "File copied successfully." << endl;

    return 0;
}
```

Output:

```
Enter the name of the source file: output.txt
Enter the name of the destination file: output2.txt
File copied successfully
```

Q3.Write a program to maintain a elementary database of employees using files.

Program:

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

class Employee {
public:
    int id;
    string name;
    int age;
    double salary;

    friend istream& operator>>(istream& is, Employee& emp);
    friend ostream& operator<<(ostream& os, const Employee& emp);
};

istream& operator>>(istream& is, Employee& emp) {
    cout << "Enter employee ID: ";
    is >> emp.id;
    is.ignore();

    cout << "Enter employee name: ";
    getline(is, emp.name);

    cout << "Enter employee age: ";
    is >> emp.age;

    cout << "Enter employee salary: ";
    is >> emp.salary;

    return is;
}

ostream& operator<<(ostream& os, const Employee& emp) {
    os << emp.id << "\t" << emp.name << "\t" << emp.age << "\t" << emp.salary << "\n";
    return os;
}

class EmployeeDatabase {
private:
    fstream file;

public:
    EmployeeDatabase(const string& filename) {
        file.open(filename, ios::in | ios::out | ios::binary | ios::app);

        if (!file) {
            cerr << "Error: Unable to open file.\n";
            exit(1);
        }
    }

    ~EmployeeDatabase() {
        file.close();
    }

    void addEmployee() {
        Employee emp;

        cin >> emp;

        file.write((char*)&emp, sizeof(emp));
        cout << "Employee added successfully.\n";
    }

    void listEmployees() {
        Employee emp;

        cout << "ID\tName\tAge\tSalary\n";
        cout << "----------------------------------------\n";

        file.seekg(0);
        while (file.read((char*)&emp, sizeof(emp))) {
            cout << emp;
        }
    }
};

int main() {
    EmployeeDatabase db("employees.dat");

    int choice;
    do {
        cout << "\nEmployee Database\n";
        cout << "1. Add employee\n";
        cout << "2. List employees\n";
        cout << "3. Exit\n";
        cout << "Enter choice: ";
        cin >> choice;

        switch (choice) {
            case 1:
                db.addEmployee();
                break;
            case 2:
                db.listEmployees();
                break;
            case 3:
                break;
            default:
                cout << "Invalid choice. Try again.\n";
                break;
        }
    } while (choice != 3);

    return 0;
}
```

Output:

```
Employee Database
1. Add employee
2. List employees
3. Exit
Enter choice: 1
Enter employee ID: 123
Enter employee name: ramesh
Enter employee age: 43
Enter employee salary: 6000000
Employee added successfully.

Employee Database
1. Add employee
2. List employees
3. Exit
Enter choice: 2
ID      Name    Age     Salary
---------------------------------------
123     ramesh      43      6000000
```

Q4.Write a program for reading and writing data to and from the file using command line arguments.

Program:

```
#include <iostream>
#include <fstream>
```

```
using namespace std;

int main(int argc,char *argv[]) {

        fstream my_file;
        my_file.open(argv[1], ios::out);
        if (!my_file) {
                cout << "File not created!"<<endl;
        }
        else {
                cout << "File created successfully!";
                my_file << "Hello World!";
                my_file.close();
        }

        //fstream my_file;
        my_file.open(argv[1], ios::in);
        if (!my_file) {
                cout << "No such file";
        }
        else {
                char ch;
                while (1) {
                        my_file >> ch;
                        if (my_file.eof())
                                break;
                        cout << ch;
                }
        }
        my_file.close();
        return 0;
}
```

my_file.txt:

```
HelloWorld!
```

Command Line Argument:

```
my_file.txt
```

Output:

```
File created successfully!
HelloWorld!
```

Q5.Write a function in C++ to count and display the number of lines not starting with alphabet 'A' present in a text file "STORY.TXT".

Program:

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

void countAndDisplay() {

  ifstream ifs("STORY.TXT");
  if (!ifs) {
    cout << "Error: Failed to open input file STORY.TXT" << endl;
    return;
  }


  int count = 0;
  string line;
  while (getline(ifs, line)) {
    if (line[0] != 'A' && line[0] != 'a') {
      count++;
      cout << line << endl;  // Display the line
    }
  }


  ifs.close();


  cout << "Number of lines not starting with 'A': " << count << endl;
}

int main() {
  countAndDisplay();
  return 0;
}
```

Output:

```
One morning, when Gregor Samsa woke from troubled dreams, he found himself transformed in his bed into a horrible vermin.
He lay on his armour-like back, and if he lifted his head a little he could see his brown belly, slightly domed and divided by arches into stiff sections. The bedding was hardly able to cover it and seemed ready to slide off any moment.
His many legs, pitifully thin compared with the size of the rest of him, waved about helplessly as he looked. "What's happened to me? " he thought. It wasn't a dream. His room, a proper human room although a little too small, lay peacefully between its four familiar walls.
It showed a lady fitted out with a fur hat and fur boa who sat upright, raising a heavy fur muff that covered the whole of her lower arm towards the viewer. Gregor then turned to look out the window at the dull weather. Drops of rain could be heard hitting the pane, which made him feel quite sad.
"How about if I sleep a little bit longer and forget all this nonsense", he thought, but that was something he was unable to do because he was used to sleeping on his right, and in his present state couldn't get into that position. However hard he threw himself onto his right, he always rolled back to where he was. He must have tried it a hundred times, shut his eyes so that he wouldn't have to look at the floundering legs, and only stopped when
Number of lines not starting with 'A': 6
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310907) OOP Laboratory

****

Class: FY-MCA Shift / Div : B Roll Number : 56

Name: Ayush Raut Assignment No : 5 Date of Implementation : 20/02/23

****

Q 1]. Write a C++ program for Exception Handling Divide by zero Using C++ Programming

Program:

```
#include<iostream>
using namespace std;
int main()
 {  int a,b;
          cout<<"Enter the value of a and b: ";
          cin>>a>>b;
           try
           {
                     if(b==0)
                    {
                              throw b;
                    }
                     else
                    {
                              cout<<"Division of a /b : "<<a/b;
                    }
           }

                     catch(int b )
                     {
                     cout<<"the denominator cannot be 0";
                     }
           }
```

Output:

```
Enter the value of a an b: 45 0
the denominator cannot be 0
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

(310907) OOP Laboratory

****

Class: FY-MCA Shift / Div : B Roll Number : 56

Name: Ayush Raut Assignment No : 5 Date of Implementation : 20/02/23

****

Q 2]. Write a C++ program to sort an array in ascending order using function template.

Program:

```
#include<iostream>
using namespace std;

template <class T>
T sort(T arr[], int n){
        int size, i, j, index;
        T temp, min;
        size = sizeof(T);

        for(i=0; i<n; i++){
                min = arr[i];
                index = i;

                for(j = i; j<n; j++){
                        if(arr[j]<min){
                                index = j;
                                min = arr[j];
                        }
                }

                temp = arr[index];
                arr[index] = arr[i];
                arr[i] = temp;
        }
}
```

```
int main(){
        int min, i, j, n;
        float arr[100];

        cout<<"\nEnter the number of element in array:";
        scanf("%d", &n);
        cout<<"\nEnter the array elements : "<<endl;
        for(i=0; i<n; i++)
                cin>>arr[i];
        sort<float>(arr, n);

        cout<<"\n Ascending order of Array Elements : ";
        for(i=0; i<n; i++)
                cout<<arr[i]<<" ";
}
```

Output:

```
Enter the number of element in array:5

Enter the array elements :
45 69 78 12 36

 Ascending order of Array Elements : 12 36 45 69 78
```

****

Class: FY-MCA Shift / Div : B Roll Number : 56

Name: Ayush Raut Assignment No : 5 Date of Implementation : 20/02/23

****

Q 3]. Write a C++ program to swap data using a function template.

Program:

```
#include<iostream>
using namespace std;

template <class T>
void swap(T *a, T *b){
	T temp;
	temp = *a;
	*a = *b;
	*b = temp;
}

int main(){
	char a, b;
	int c , d;
	cout<<"\nEnter the value of a and b : ";
	cin>>a>>b;
	cout<<"\nBefore Swap value of a = "<<a<<" b = "<<b<<endl;
	swap<char>(&a, &b);
	cout<<"\nAfter Swap value of a = "<<a<<" b = "<<b<<endl;
	cout<<"\nEnter the value of c and d : ";
	cin>>c>>d;
	cout<<"\nBefore Swap value of c = "<<c<<" d = "<<d<<endl;
	swap<int>(&c, &d);
	cout<<"\nAfter Swap value of c = "<<c<<" d = "<<d<<endl;
	return 0;
}
```

Class: FY-MCA Shift / Div : B Roll Number : 56

Name: Ayush Raut Assignment No : 5 Date of Implementation : 20/02/23

Output:

```
Enter the value of a and b : X Y

Before Swap value of a = X b = Y

After Swap value of a = Y b = X

Enter the value of c and d : 5 8

Before Swap value of c = 5 d = 8

After Swap value of c = 8 d = 5
```

(310907) OOP Laboratory

****

Class: FY-MCA   Shift / Div : B   Roll Number : 56

Name: Ayush Raut   Assignment No : 5   Date of Implementation : 20/02/23

****

Q 4]. Write a C++ program to make a simple calculator using a class template.

Program:

```
#include<iostream>
using namespace std;

template <class T>
class Calculator{
        protected:
                T a, b;
        public:
                Calculator(){
                        cout<<"\nEnter the value of a and b : ";
                        cin>>a>>b;
                }
                void display(){
                        cout<<"\nA = "<<a<<"\nB = "<<b;
                }
                inline T add(){
                        return (a+b);
                }
                inline T sub(){
                        return (a-b);
                }
                inline T div(){
                        return (a/b);
                }
                inline T mul(){
                        return (a*b);
                }
};
```

```
int main(){
        Calculator<float>c;
        c.display();
        int ch;
        label1:

        cout<<"\n1.Addition"<<endl;
        cout<<"2.Subtraction"<<endl;
        cout<<"3.Division"<<endl;
        cout<<"4.Multiplication"<<endl;
        cout<<"\nEnter you choice : "<<endl;
        cin>>ch;
        switch(ch){
                case 1:
                        cout<<"Addition = "<<c.add();
                        break;
                case 2:
                        cout<<"Subtraction = "<<c.sub();
                        break;
                case 3:
                        cout<<"Division = "<<c.div();
                        break;
                case 4:
                        cout<<"Multiplication = "<<c.mul();
                        break;
                default:
                        cout<<"\nEnter the valid choice.";
                        goto label1;
        }
        return 0;
}
```

Output:

```
Enter the value of a and b : 45 69

A = 45
B = 69
1.Addition
2.Subtraction
3.Division
4.Multiplication

Enter you choice :
1
Addition = 114
```

Q 5]. Write a C++ program implementing stack and its operations using template class.

Program:

```
#include<iostream>
#define MAX 5
using namespace std;

template <class T>
class Stack{
        protected:
                T arr[MAX];
                int top;
        public:
                Stack(){
```

```
            cout<<"Stack Created Successfully"<<endl;
            top = -1;
        }
        void push(){
            T a;
            if(isFull())
                cout<<"Stack is full."<<endl;
            else{
                cout<<"Enter the element want to push : "; cin>>a;
                arr[++top]=a;
            }
        }

        void pop(){
            T a;
            if(isEmpty())
                cout<<"Stack is Empty"<<endl;
            else{
                a = arr[top--];
                cout<<"Popped Element is "<<a<<endl;
            }
        }

        int isFull(){
            if(top == MAX-1)
                return 1;
            else
                return 0;
        }

        int isEmpty(){
            if(top==-1)
                return 1;
            else
                return 0;
        }

        void display(){
            for(int i = 0; i<=top; i++)
                cout<<" "<<arr[i];
        }
};

int main(){
    int ch;
    Stack<int>s;
    do{
        cout<<"\n1.Push\n2.Pop\n3.isFull\n4.isEmpty\n5.Display\n6.Exit\n Enter your choice : ";
        cin>>ch;

        switch(ch){
            case 1:
                s.push();
                break;
            case 2:
                s.pop();
                break;
            case 3:
                if(s.isFull())
                    cout<<"\nStack is full"<<endl;
                else
                    cout<<"Stack is not full"<<endl;
                break;
            case 4:
                if(s.isEmpty())
                    cout<<"\nStack is Empty"<<endl;
                else
                    cout<<"Stack is not Empty"<<endl;
                break;
            case 5:
                s.display();
                break;
            case 6:
                break;
            default :
                cout<<"Enter correct Choice."<<endl;
                break;
        }
    }while(ch!=6);
}
```

Output:

```
Stack Created Successfully

1.Push
2.Pop
```

```
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 1
Enter the element want to push : 56

1.Push
2.Pop
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 1
Enter the element want to push : 39

1.Push
2.Pop
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 1
Enter the element want to push : 48

1.Push
2.Pop
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 5
 56 39 48
1.Push
2.Pop
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 2
Popped Element is 48
```

```
1.Push
2.Pop
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 3
Stack is not full

1.Push
2.Pop
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 4
Stack is not Empty

1.Push
2.Pop
3.isFull
4.isEmpty
5.Display
6.Exit
 Enter your choice : 6
```

Q 6]. Write a C++ program implementing linked list & some required operations on it using class template

Program:

```
#include<iostream>
#include<stdlib.h>
using namespace std;

template<class T>
class Node{
        public:
                T data;
                Node *next;
};
```

```
template <class T>
class LinkedList{
    public:
        Node<T> *start;
    public:
        LinkedList(){
            start = NULL;
        }
        void createList(){
            Node<T> *temp, *newnode;
            int n, i;
            cout<<"\nEnter the number of nodes you want : ";
            cin>>n;
            temp = start;
            for(i=0; i<n; i++){
                newnode = createNode();
                if(start == NULL)
                    temp = start = newnode;
                else{
                    temp->next = newnode;
                    temp = temp->next;
                }
            }
        }

        void display(){
            Node<T> *temp;
            cout<<endl;
            cout<<"\nLinked List = ";
            for(temp = start; temp!=NULL;temp = temp->next)
                cout<<temp->data<<" ";
        }

        Node<T> *createNode(){
            Node<T> *newnode;
            newnode = (Node<T>*)malloc(sizeof(Node<T>));
            cout<<"Enter the data : ";
            cin>>newnode->data;
            newnode->next = NULL;
            return newnode;
        }

        void insertBeg(){
            Node<T> *newnode;
            newnode = createNode();
            if(start == NULL)
                start = newnode;
            else{
                newnode->next = start;
                start = newnode;
            }
        }

        void insertEnd(){
            Node<T> *newnode, *temp;
            newnode = createNode();
            if(start == NULL)
                start = newnode;
            else{
                for(temp=start; temp->next!=NULL; temp = temp->next);
                temp->next = newnode;
            }
        }

        void insertMid(){
           int n;
            int i;
            Node<T> *newnode, *temp;
            cout<<"\nEnter the Position to Insert: ";
            cin>>n;
            newnode = createNode();
            if(n==1){
              newnode->next = start;
              start = newnode;
            }
            else{
                temp = start;
                for(i=0, temp = start; (i<n-2) && (temp->next!=NULL); temp =
temp->next, i++);
                newnode->next = temp->next;
                temp->next = newnode;
            }
        }

        void deleteBeg(){
            Node<T> *temp, *temp1;
            if(start == NULL)
```

```
                                cout<<"\nList is Already Empty."<<endl;
                        else{
                                temp = start;
                                start = start->next;
                                temp->next = NULL;
                                free(temp);
                        }
                }

                void deleteEnd(){
                        Node<T> *temp, *temp1;
                        for(temp = start; temp->next->next!=NULL; temp = temp->next);
                        temp1 = temp->next;
                        temp->next = NULL;
                        free(temp);
                }

                void deletePos(){
                        Node<T> *temp, *temp1, *temp2;
                        int n, i;
                        if(start == NULL){
                          cout<<"\nLinked List is Null.";
                        }
                        else{
                          cout<<"\nEnter the position to be deleted : ";
                          cin>>n;
                          for(i=0, temp = start; (i<n-2)&&(temp!=NULL); temp = temp->next, i++);
                          temp1 = temp->next;
                          temp2 = temp1->next;
                          temp->next = temp2;
                          temp1->next = NULL;
                          free(temp1);
                        }
                }
};
int main(){
        LinkedList<int> ls;
        int ch, ch2;
        do{
          cout<<"\n1.CreateList.\n2.Insert.\n3.Delete.\n4.Display\n Enter your choice : ";
          cin>>ch;
          switch(ch){
            case 1:
              ls.createList();
```

```
 break;
case 2:
   cout<<"\n1.Insert at Beg.\n2.Insert at Mid.\n3.Insert at End."<<endl;
   cout<<"Enter your choice : "; cin>>ch2;

   switch(ch2){
       case 1:
                  ls.insertBeg();
                  break;
             case 2:
                  ls.insertMid();
                  break;
             case 3:
                  ls.insertEnd();
                  break;
             default:
                  cout<<"\nEnter the correct choice . "<<endl;
                  break;
     }
   break;

case 3:
   cout<<"\n1.Delete at Beg. \n2.Delete at Mid. \n3.Delete at End."<<endl;
   cout<<"Enter your choice : "; cin>>ch2;

   switch(ch2){
             case 1:
                  ls.deleteBeg();
                  break;
             case 2:
                  ls.deletePos();
                  break;
        case 3:
        ls.deleteEnd();
        break;
        default:
        cout<<"\nEnter the correct choice . "<<endl;
        }
   break;

case 4:
 ls.display();
 break;
```

```
            case 5:
              break;
          }
        }while(ch!=5);
        return 0;
}
```

Output:

```
1.CreateList.
2.Insert.
3.Delete.
4.Display
 Enter your choice : 1

Enter the number of nodes you want : 5
Enter the data : 46
Enter the data : 12
Enter the data : 36
Enter the data : 58
Enter the data : 97

1.CreateList.
2.Insert.
3.Delete.
4.Display
 Enter your choice : 2

1.Insert at Beg.
2.Insert at Mid.
3.Insert at End.
Enter your choice : 1
Enter the data : 26

1.CreateList.
2.Insert.
3.Delete.
4.Display
 Enter your choice : 4
```

```
Linked List = 26 46 12 36 58 97
1.CreateList.
2.Insert.
3.Delete.
4.Display
 Enter your choice : 3

1.Delete at Beg.
2.Delete at Mid.
3.Delete at End.
Enter your choice : 1

1.CreateList.
2.Insert.
3.Delete.
4.Display
 Enter your choice : 4


Linked List = 46 12 36 58 97
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

****

1. Define a class animal with their basic features as class members .create two derive classes from animal named herbivorous and carnivorous (type) with their own features too. accept name of animal with type and display all the related information.

```
#include<iostream.h>
#include<conio.h>
class Animal
{
  char name[30];
  char type[20];
  public:
       void getdata()
       {
          cout<<"Enter the  name of an Animal:";
          cin.getline(name,30);
          cout<<endl;
          cout<<"Enter the type of an Animal:";
          cin.getline(type,20);
       }
       void putdata()
       {
           cout<<name<<" "<<type<<endl;
       }
 };
class Herbivores : private Animal
{
      char nm[50];
      char food[50];

      public:
             void getdata()
             {
               cout<<"Enter the name of Herbivores animal:";
               cin.getline(nm,50);
               cout<<endl;
               cout<<"Enter the food of Herbivores Animal:";
               cin.getline(food,50);

            }
            void putdata()
            {
               cout<<endl;
               cout<<nm<<" "<<food;
            }
};
class Carnivores:private Animal
{
```

```
             char nm[50];
             char food[50];
             public:
                void getdata()
                {
                   cout<<"Enter the name of the carnivores animal:";
                   cin.getline(nm,50);
                   cout<<endl;
                   cout<<"Enter the food of the carnivores animal:";
                   cin.getline(food,50);
                   cout<<endl;
                }
                void putdata()
                {
                  cout<<nm<<" "<<food;
                }
 };
void main()
{
       clrscr();
       Herbivores h1;
       h1.getdata();

       Carnivores c1;
       c1.getdata();
       cout<<endl;
       h1.putdata();
       cout<<endl;
       c1.putdata();
       getch();
 }
```

Output:

```
Enter the name of Herbivores animal: deer
Enter the food of Herbivores animal: plant
Enter the Carnivores animal: tiger
Enter the food of the carnivores animal: meat

Deer plant
Tiger meat
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

****

2. Write a program to derive a class rectangle from base class shape using single inheritance.

```
#include<iostream>
 using namespace std;
 class shape
 {
            public:
                        float length,breadth;
            public:
                void getdata()
                        {
cout<<"enter the length:";
                                    cin>>length;
                                    cout<<"enter the breadth:";
                                    cin>>breadth;
                                    }
};
 class rectangle:public shape

 {
            private:
                float area;
            public:
                        void calculate()
                        {
                                    area=length*breadth;
                                    cout<<"area of rectangle is"<<area;
                        }
};
```

```
int main()
{
        rectangle r;
        r.getdata();
        r.calculate();
        return 0;
}
```

Output:

```
enter the length:3
enter the breadth:5
area of rectangle is15
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

****

3. Define a class containing operator function to overload unary minus ('-') operator.

```
#include<iostream.h>
#include<conio.h>
class demo
{
      private:
            int a,b,c;
      public:
            void getdata()
            {
                   cout<<"enter the value of a:";
                   cin>>a;
                   cout<<"enter the value of b:";
                   cin>>b;
                   cout<<"enter the value of c:";
                   cin>>c;
             }
             void operator-()
             {
                  a=-a;
                  b=-b;
                  c=-c;
            }
            void showdata()
            {
                  cout<<"value of a is"<<a;
                  cout<<"\nvalue of b is"<<b;
                  cout<<"\nvalue of e is"<<c;
            }
};
void main()
```

```
{
        demo d;
        clrscr();
        d.getdata();
        -d;
        d.showdata();
        getch();
}
```

Output:

```
enter the value  of a:6
enter the value of b:7
enter the value of c:9
value of a is-6
value of b is-7
value of c is-9
```

A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

****

4. Define base class students with roll number name marks 1 marks to marks 3 AS data members defence sport class with sport name participation level achievement as data member and also find sports grade in a rate student class in sports class to find final grade of 5 students (District/ Gold-3 District/Silver -2 District/Bronze-1 State/Gold-6 state/Silver-5 State/Bronze-4 National/Gold-9 National/Silver -8 National/Bronze-7)

Program:

```
#include<iostream>
using namespace std;
class stud_info{
protected:
int roll;
char name[30];
public:
void get_stud_data(){
cout<<"Enter your roll Number: ";
cin>>roll;
cout<<"Enter your name: ";
cin>>name;
}
};
class test_info:virtual public stud_info{
protected:
int test1,test2,test3;
public:
void get_test_data(){
cout<<"Enter marks of first test: ";
cin>>test1;
cout<<"Enter marks of second test: ";
cin>>test2;
cout<<"Enter the marks of third test: ";
cin>>test3;
}
};
class sports_info : virtual public stud_info{
protected:
char sports_remark[30];
int sports_mark;
public:
void get_sports_marks(){
cout<<"Enter your sports remark: ";
cin>>sports_remark;
if(sports_remark=="dist gold"){
sports_mark=3;
}
else if(sports_remark=="dist silver"){
sports_mark=2;
}
else if(sports_remark=="dist bronze"){
```

```
sports_mark=1;
}
else if(sports_remark=="state g"){
sports_mark=6;
}
else if(sports_remark=="state s"){
sports_mark=5;
}
else if(sports_remark=="state b"){
sports_mark=4;
}
else if(sports_remark=="national g"){
sports_mark=9;
}
else if(sports_remark=="national s"){
sports_mark=8;
}
else if(sports_remark=="national b"){
sports_mark=7;
}
}
};
class result : public test_info,public sports_info{
protected:
int per;
public:
void cal(){
per=per+sports_mark;
per=(test1+test2+test3)/3;
}
void display(){
cout<<"Roll number: "<<roll<<"\nName: "<<name<<"\nYour percentage: "<<per<<"%";
}
};
int main(){
result obj;
obj.get_stud_data();
obj.get_test_data();
obj.get_sports_marks();
obj.cal();
obj.display();
return -1;
}
```

Output:

```
Enter your roll Number: 51156
Enter your name: Ayush
Enter marks of first test: 89
Enter marks of second test: 90
Enter the marks of third test: 91
Enter your sports remark: Gold
Roll number: 51156
Name: Ayush
Your percentage: 90%
```

MCA Department
A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

****

5. Write c++ program to read the best class information i.e. employees, name code, designation .and the derived class contains year of experience and age. Design a virtual based class for the item, employee name and code.

```
#include<iostream.h>
#include<conio.h>
class Employee
{
 public:
 char Emp_name[50];
 int Emp_code;
 char designation[50];
 public:
 void getdata()
 {
 cout<<"Enter the employee name:";
 cin>>Emp_name;
 cout<<"Enter the Emp_code:";
 cin>>Emp_code;
 cout<<"Enter the designation of employee:";
 cin>>designation;
 }
};
class Information :public virtual Employee
{
 int Experience;
 int age;
 public:
 void putdata()
 {
 cout<<"Enter the years of experience:";
 cin>>Experience;
 cout<<"Enter the age of employee:"<<endl;
 cin>>age;
 }
 void show()
 {
cout<<"Employee Name="<<Emp_name<<endl;
 cout<<"Emplpyee code="<<Emp_code<<endl;
 cout<< "Employee designation="<<designation<<endl;
cout<<"Employee Experience="<<Experience<<endl;
 cout<<"Employee age="<<age;
 }
};
void main()
{
clrscr();
 Information obj;
 obj.getdata();
 obj.putdata();
 obj.show();
 getch();
}
```

Output:

```
 Enter the employee name:ayush
Enter the Emp_code:123
Enter the designation of employee:Manager
Enter the years of experience:5
Enter the age of employee:30
Employee Name=ayush
Employee code=123
Employee designation=Manager
Enter the years of experience:5
Enter the age of employee:30
Employee Name=ayush
Employee code=123
Employee designation=Manager
Employee Experience=5
Employee age=30
```

A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

6. Write a program of multiple inheritance for a company that publishes and markets books. Derive a pamphlet a book and tape from publication and tape and book from cells and notice from pamphlet display a book information.

```
#include <iostream>
using namespace std;
class pamphlet
{
   public:
      void showp()
      {
         cout<<"The name of the book is THE WINGS OF FIRE";
      }
};
class book
{
   public:
      void showb()
      {
         cout<<"\nIt is based on the life of APJ ABDUL KALAM.";
      }
};
class tape
{
   public:
      void showt()
      {
         cout<<"\nfirst edition of the book is released on 1999";
      }
};
class publication:public pamphlet,public book,public tape
{
   public:
      void pub()
      {
         cout<<"\nIt is published by UNIVERSITIES PRESS";
      }
};
class sales:public tape,public book
{
   public:
      void showsale()
      {
         cout<<"\nThe price of the book is 200.";
      }
};
int main()
{
   publication p;
   p.showp();
```

```
    p.showb();
    p.showt();
    p.pub();
    cout<<"\n------------------------------------------------------------------------------";
    sales s;
    s.showsale();
    s.showb();
    s.showt();
    return 0;
}
```

Output:

```
The name of the book is THE WINGS OF FIRE
It is based on the life of APJ ABDUL KALAM.
first edition of the book is released on 1999
It is published by UNIVERSITIES PRESS
------------------------------------------------------------------------------
The price of the book is 200.
It is based on the life of APJ ABDUL KALAM.
first edition of the book is released on 1999
```

MCA Department
A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

Q7. Write a class 'point' with x and y coordinates as data members. Derive two classes 'Line' and 'Circle' from 'point' with appropriate data members. Derive a class 'triangle' from 'line'. Implement read() and draw() member functions for all the above classes.

```
#include<iostream>
using namespace std;
class Triangle
{
   public:
      void T()
      {
         cout<<"class Triangle Derived from class line\n";
         cout<<"draw Triangle";
      }

};
class line:public Triangle
{
   int x;
   public:
      void read()
      {
         cout<<"\nenter centimeter:";
         cin>>x;
      }
      void draw()
      {
         cout<<"draw line\n";
      }
};
class circle
{
   float r;
   public:
      void read()
      {
         cout<<"\nenter radius";
         cin>>r;
      }
      void draw()
      {
         cout<<"draw circle\n";
      }
};
class point:public line,public circle
{
   int x,y;
   public:
        void read()
        {
            cout<<"\nenter the value of x:";
            cin>>x;
            cout<<"\nenter the value of y:";
            cin>>y;
        }
        void draw()
        {
            cout<<"draw point\n";
        }
};

int main()
{
    point p;
    line l;
    circle c;
    p.read();
    p.draw();
    l.read();
    l.draw();
    l.T();
    c.read();
    c.draw();
    return 0;

}
```

Output:

```
enter the value of x:3

enter the value of y:4
draw point

enter centimeter:6
draw line
class Triangle Derived from class line
draw Triangle
enter radius8
draw circle
```

A.Y.2022-23

****

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

Q8. .Create a class FLOAT that contains one float data member. Overload all the four arithmetic operators so that they operate on the objects of FLOAT.

Program:

```
 #include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class FLOAT
{
 float x,a,b,c,d,f;
 public:
 FLOAT()
 {
 x=20;
 }
 FLOAT(float i)
 {
 x=i;
 }
 void operator +(FLOAT);
 void operator -(FLOAT);
 void operator *(FLOAT);
 void operator /(FLOAT);
};
void FLOAT::operator +(FLOAT f)
{
a=x+f.x;
cout<<"\nAddition is:"<<a;
}
void FLOAT::operator -(FLOAT f)
{
b=x-f.x;
cout<<"\nSubtraction is:"<<b;
}
void FLOAT::operator *(FLOAT f)
{
c=x*f.x;
cout<<"\nMultiplication is:"<<c;
}
void FLOAT::operator/(FLOAT f)
{
	d=x/f.x;
cout<<"\nDivision is:"<<d;
}
```

```
void main()
{
clrscr();
FLOAT F1;
FLOAT F2(100);
FLOAT F3;
F1+F2;
F1-F2;
F1*F2;
F1/F2;
getch();
}
```

Output:

```
 Addition is:120
Subtraction is:-80
Multiplication is:2000
Division is:0.2
```

Q9. Create a class MAT of size m*n. Define all possible matrix operations for MAT type objects.

```
MAT M3;
 for(int i=0;i<2;i++)
 {
 for(int j=0;j<2;j++)
 {
 M3.a[i][j]=0;
 for(int k=0;k<2;k++)
 {
 M3.a[i][j]=(a[i][k]*M2.a[k][j])+M3.a[i][j];
 }
 }
 }
 return M3;
 }
} ;
int main()
{
 clrscr();
MAT M1,M2,M3;
cout<<"\n Enter Matrix M1 value:";
M1.accept();
cout<<"\n Enter Matrix M2 value:";
M2.accept();
M3=M1+M2;
cout<<"\n Addition of M1+M2:";
M3.display();
M3=M1*M2;
cout<<"\n Multiplication of M1*M2:";
M3.display();
getch();
return 0;
}
```

Output:

```
 Enter Matrix M1 value:
Enter 4 element:1 2 3 4
Enter Matrix M2 value:
Enter 4 element: 5 4 6 3
```

```
Addition of M1+M2:
6 6
9 7
Multiplication of M1*M2:
17 10
39 24
```

Class: FY-MCA Shift / Div : B Batch: F3 Roll Number : 51156

Name:Ayush Kishor Raut Assignment No : 3 Date of Implementation :

Q10. Write a c++ program to concatenate two strings using operator + function.

Program:

```
#include <iostream>
#include <string.h>
 using namespace std;
class AddString {
 public:
   char s1[25], s2[25];
AddString(char str1[], char str2[])
    {
        strcpy(this->s1, str1);
        strcpy(this->s2, str2);
    }
    void operator+()
    {
        cout << "\nConcatenation: " << strcat(s1, s2);
    }
};
int main()
{
    char str1[] = "hello";
    char str2[] = "world";
 AddString a1(str1, str2);
  +a1;
  return 0;
}
```

Output:

```
Concatenation: helloworld
```

Progressive Education Society's

Modern College of Engineering, Pune

MCA Department

A.Y.2022-23

Object oriented programming

****

Class: FY-MCA　Shift / Div : B　Roll Number : 56

Name: Ayush Kishor Raut　Assignment No : 4　Date of Implementation : 21-02-2023

****

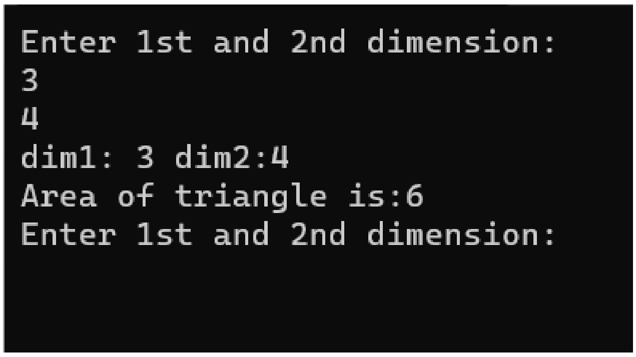
Q1. Create a base class shape. Derive two classes as Triangle and Rectangle from the base class shape. Take getdata() and display_area() as member functions of base class. Make display_area() as a virtual function and redefine it in derived classes to suit their requirement. Design a program that will accept dimensions of a triangle or rectangle interactively and display the area.

Program:-

```
#include <iostream>
using namespace std;

class Shape
{
        protected:
                float dim1,dim2;
        public:
                void get_data(float x,float y)
                {
                        dim1=x;
                        dim2=y;
                }

                virtual void display_area(){ };
};
```

```
class Rectangle:public Shape
{
        public:
                void display_area()
                {
                        float rec;
                        cout<<"dim1: "<<dim1<<" "<<"dim2:" <<dim2<<endl;
                        rec=dim1*dim2;
                        cout<<"Area of rectangle is:"<<rec<<endl;
                }
};

class Triangle:public Shape
{
        public:
                void display_area()
                {
                        float tri;
                        cout<<"dim1: "<<dim1<<" "<<"dim2:" <<dim2<<endl;
                        tri=0.5*dim1*dim2;
                        cout<<"Area of triangle is:"<<tri<<endl;
                }
};

int main(void)
{
        Shape *s;
        float d1,d2;
        Triangle t;
        s=&t;
        cout<<"Enter 1st and 2nd dimension: "<<endl;
        cin>>d1>>d2;
        s->get_data(d1,d2);
        s->display_area();
        Rectangle r;
        s=&r;
        cout<<"Enter 1st and 2nd dimension: "<<endl;
        cin>>d1>>d2;
        s->get_data(d1,d2);
        s->display_area();

        return 0;
}
```

OUTPUT:-



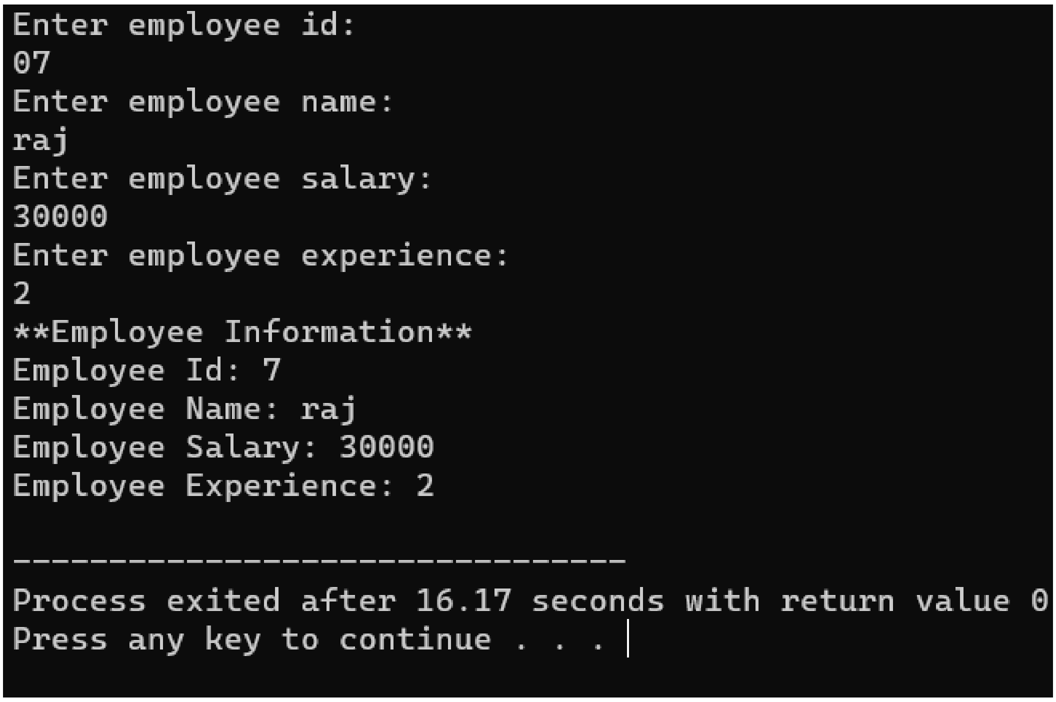
```
Enter 1st and 2nd dimension:
3
4
dim1: 3 dim2:4
Area of triangle is:6
Enter 1st and 2nd dimension:
```

Q2. Write a C++ program to demonstrate a pure virtual function which is invoked from the object of derived class through the pointer of the base class. Base class contains getdata() and display(). Display the information of employee using this.

Program:- #include <iostream>

```
using namespace std;

class Base
{
        public:
                void accept();
                virtual void display()=0;  //pure virtual function
};
```

```
class Derived:public Base
{
        int id,exp;
        string name;
        float sal;
        public:
                void accept()
                {
                        cout<<"Enter employee id: "<<endl;
                        cin>>id;
                        cout<<"Enter employee name: "<<endl;
                        cin>>name;
                        cout<<"Enter employee salary: "<<endl;
                        cin>>sal;
                        cout<<"Enter employee experience: "<<endl;
                        cin>>exp;
                }
                void display()
                {
                        cout<<"**Employee Information**"<<endl;
                        cout<<"Employee Id: "<<this->id<<endl;
                        cout<<"Employee Name: "<<this->name<<endl;
                        cout<<"Employee Salary: "<<this->sal<<endl;
                        cout<<"Employee Experience: "<<this->exp<<endl;
                }
};

int main(void)
{
        Base *b;
        Derived d;
        b=&d;
        d.accept();
        b->display();
        return 0;
}
```



```
Enter employee id:
07
Enter employee name:
raj
Enter employee salary:
30000
Enter employee experience:
2
**Employee Information**
Employee Id: 7
Employee Name: raj
Employee Salary: 30000
Employee Experience: 2

--------------------------------
Process exited after 16.17 seconds with return value 0
Press any key to continue . . .
```

Q3. Consider a book shop which sales books and video tapes. Define a class Media which store title and price of a publication. Create two derive classes book and tape where book class is used to store no of pages of book and tape class is used to store playing time of a tape. Use display function in all the classes and show the use of virtual function.

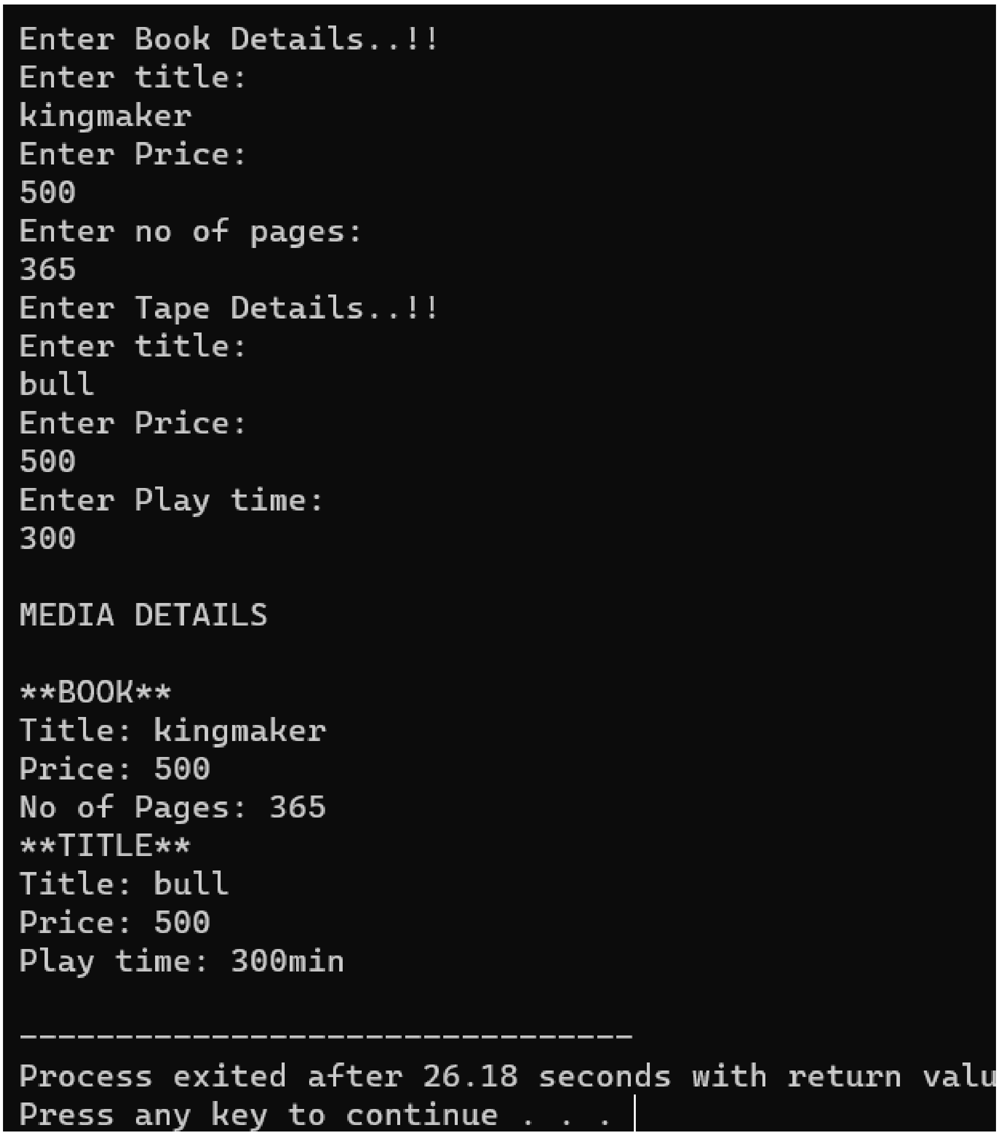
Program:-

```
#include <iostream>
#include <string>
using namespace std;

class Media
{
        protected:
                string title;
                float price;
        public:
                Media(string t,float p)
                {
```

```
            //strcpy(title,t);
            title=t;
            price=p;
        }
        virtual void display(){
          cout<<"Media class : "<<endl;
        }
};

class Book:public Media
{
    int noofpages;
    public:
        Book(string t,float p,int n):Media(t,p)
        {
            noofpages=n;
        }
        void display()
        {
            cout<<"Title: "<<title<<endl;
            cout<<"Price: "<<price<<endl;
            cout<<"No of Pages: "<<noofpages<<endl;
        }
};

class Tape:public Media
{
    float play_time;
    public:
        Tape(string t,float p,float pt):Media(t,p)
        {
            play_time=pt;
        }
        void display()
        {
            cout<<"Title: "<<title<<endl;
            cout<<"Price: "<<price<<endl;
            cout<<"Play time: "<<play_time<<"min"<<endl;
        }
};

int main(void)
{
    string title;
    int noofpages;
    float price,playtime;

    cout<<"Enter Book Details..!!"<<endl;
    cout<<"Enter title: "<<endl;
    cin>>title;
    cout<<"Enter Price: "<<endl;
    cin>>price;
    cout<<"Enter no of pages: "<<endl;
    cin>>noofpages;
    Book b(title,price,noofpages);
    cout<<"Enter Tape Details..!!"<<endl;
    cout<<"Enter title: "<<endl;
    cin>>title;
    cout<<"Enter Price: "<<endl;
    cin>>price;
    cout<<"Enter Play time: "<<endl;
    cin>>playtime;
    Tape t(title,price,playtime);

    Media *m[2];
    m[0]=&b;
    m[1]=&t;

    cout<<"\nMEDIA DETAILS\n "<<endl;
    cout<<"**BOOK**"<<endl;
```

```
        m[0]->display();

        cout<<"**TITLE**"<<endl;
        m[1]->display();
        return 0;

}
```

OUTPUT

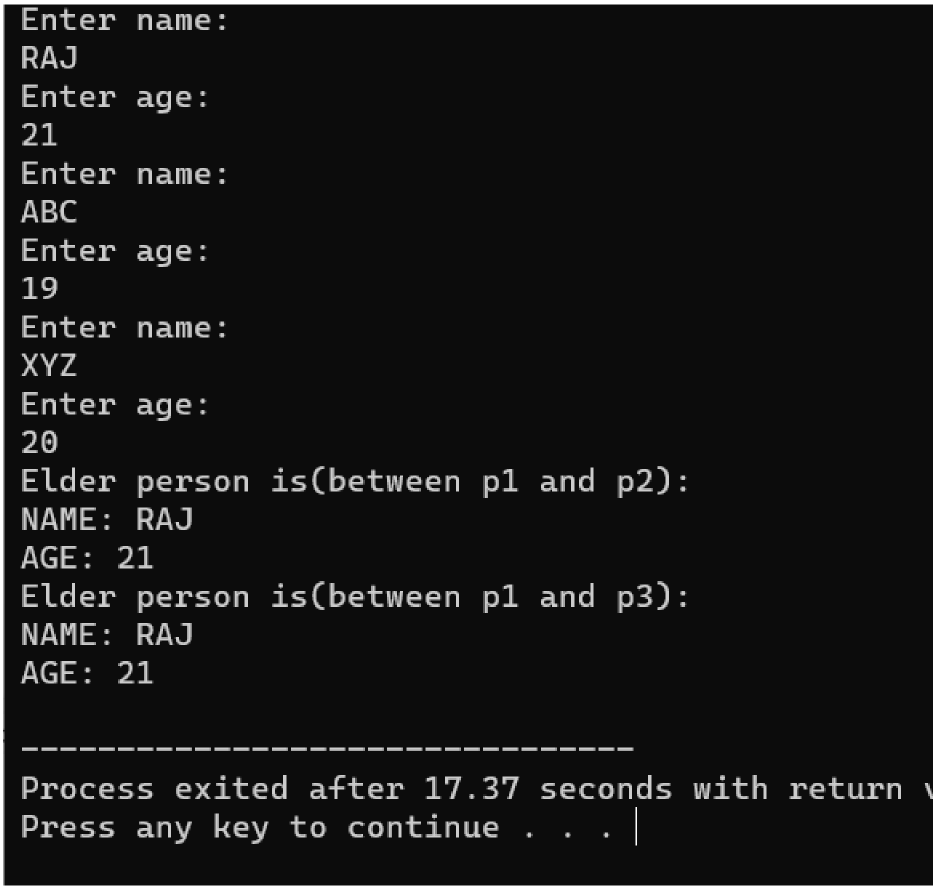


```
Enter Book Details..!!
Enter title:
kingmaker
Enter Price:
500
Enter no of pages:
365
Enter Tape Details..!!
Enter title:
bull
Enter Price:
500
Enter Play time:
300

MEDIA DETAILS

**BOOK**
Title: kingmaker
Price: 500
No of Pages: 365
**TITLE**
Title: bull
Price: 500
Play time: 300min

--------------------------------
Process exited after 26.18 seconds with return valu
Press any key to continue . . .
```

Q4:- Write a C++ program to maintain the records of person with details (Name and Age) and find the eldest among them. The program must use this pointer to return the result.

Program:-

```
#include <iostream>
using namespace std;

class Person
{
        protected:
                string name;
                int age;
        public:
                void accept()
                {
                        cout<<"Enter name: "<<endl;
                        cin>>name;
                        cout<<"Enter age: "<<endl;
                        cin>>age;
                }
                void display()
                {
                        cout<<"NAME: "<<name<<endl;
                        cout<<"AGE: "<<age<<endl;
                }

                Person greater(Person &p)
                {
                        if(p.age >= this->age)
                        return p;
                        return *this;
                }
};

int main(void)
{
        Person p1,p2,p3;
        p1.accept();
        p2.accept();
        p3.accept();
        Person p=p1.greater(p2);
        cout<<"Elder person is(between p1 and p2): "<<endl;
        p.display();

        p=p1.greater(p3);
        cout<<"Elder person is(between p1 and p3): "<<endl;
        p.display();
        return 0;
}
```

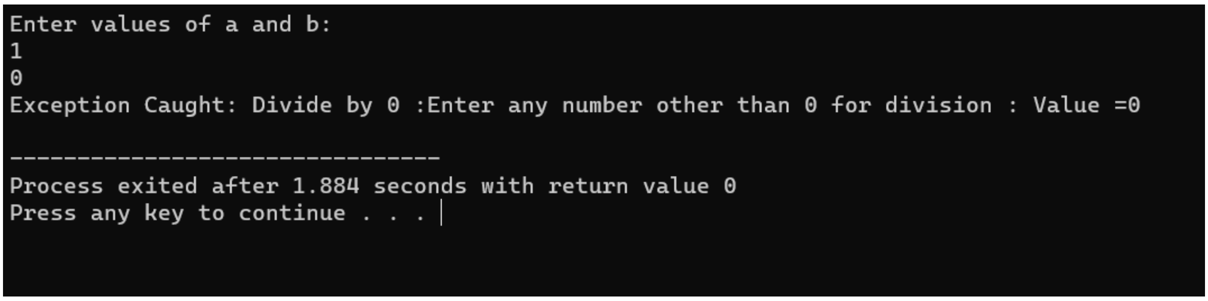


```
Enter name:
RAJ
Enter age:
21
Enter name:
ABC
Enter age:
19
Enter name:
XYZ
Enter age:
20
Elder person is(between p1 and p2):
NAME: RAJ
AGE: 21
Elder person is(between p1 and p3):
NAME: RAJ
AGE: 21

--------------------------------
Process exited after 17.37 seconds with return v
Press any key to continue . . .
```

Q5. Create a class ‘Account’ that stores customer name, account number and type of account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member

```
#include <iostream>
using namespace std;
int division(int a,int b)
{
	if(b!=0)
		cout<<a<<" / "<<b<<" = "<< a/b <<endl;
	else
		throw(b);
}

int main(void)
{
	int a,b;
	try
	{
		cout<<"Enter values of a and b: "<<endl;
		cin>> a >> b;
		division(a,b);
	}
	catch(int n)
	{
		cout<<"Exception Caught: Divide by 0 :Enter any number other than 0 for division : 
Value ="<<n<<endl;
	}
	return 0;
}
```

OUTPUT:-



```
Enter values of a and b:
1
0
Exception Caught: Divide by 0 :Enter any number other than 0 for division : Value =0

--------------------------------
Process exited after 1.884 seconds with return value 0
Press any key to continue . . .
```