**ASSIGNMENT 2**

NAME: SHUBHAM V. TAKANKHAR

CLASS SY-MCA

ROLL NO. 54

GR. NO. 119C0046

Q1: Write a program with class Complex containing data members m\_real and m\_imag. and Overload its operator using binary ‘+’ operator.

🡺

**<SOURCE\_CODE>**

#include<iostream>

using namespace std;

class Complex{

private:

int m\_real;

int m\_imag;

public:

Complex(int i=0,int r=0){

m\_real=r;

m\_imag=i;

}

void inputValues(){

cout<<"Enter Real Number: ";

cin>>m\_real;

cout<<"Enter Imaginary Number: ";

cin>>m\_imag;

}

Complex operator+(Complex obj){

Complex temp;

temp.m\_real=m\_real+obj.m\_real;

temp.m\_imag=m\_imag+obj.m\_imag;

return temp;

}

void printValues(){

cout<<"\nOUTPUT: "<<m\_real<<"+"<<m\_imag;

}

};

int main(){

Complex a,b,c;

a.inputValues();

b.inputValues();

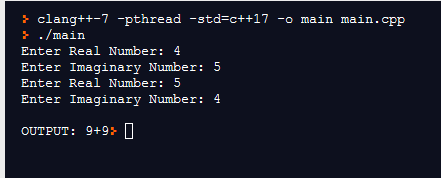
c=a+b;

c.printValues();

}

**<<OUTPUT\_SCREENS>>**

**------------------------------------------------------------------------------**



------------------------------------------------------------------------------

Q2.Write a program with class matrix and perform matrix subtraction and matrix multiplication using operator overloading.

🡺

**<SOURCE\_CODE>**

#include<iostream>

using namespace std;

class matrix{

public:

int m[3][3];

matrix(){

for (int i=0;i<3;i++){

for(int j=0;j<3;j++){

m[i][j]=0;

}

}

}

void setMatrix(){

for (int i=0;i<3;i++){

for(int j=0;j<3;j++){

cout<<"Enter elment at i:"<<i<<" and j:"<<j<<" =>";

cin>>m[i][j];

}

}

}

void printMatrix(){

for (int i=0;i<3;i++){

for(int j=0;j<3;j++){

cout<<m[i][j]<<",";

}

cout<<"\n";

}

}

matrix operator \*(matrix obj){

matrix temp;

for (int i=0;i<3;i++){

for(int j=0;j<3;j++){

  temp.m[i][j]=0;

  for(int k=0;k<3;k++){

temp.m[i][j]=temp.m[i][j]+m[j][k]\* obj.m[k][j];

  }

}

}

return temp;

}

};

int main()

{

matrix a,b,c;

a.setMatrix();

b.setMatrix();

c=a\*b;

cout<<"MATRIX A: \n";

a.printMatrix();

cout<<"\nMATRIX B: \n";

b.printMatrix();

cout<<"\nRESULT MATRIX C: \n";

c.printMatrix();

}

**<<OUTPUT\_SCREENS>>**

