**1.** Following is a declaration for a class to represent Complex numbers. A Complex number has two parts, the real part (let’s say a) and the imaginary part (say b), and is represented as a+bi where i has a value of sqrt (‐1). Write the implementation of the class.

**class complex**

**{**

**private: float real; // Real Part**

**float imag; // Imaginary Part public:**

**complex(float,float); //2-arg constructor with default parameters**

**complex add(complex);**

**complex subtract(complex);**

**complex multiply(complex);**

**complex divide(complex);**

**complex getconjugate();**

**void setdata(float,float); //assigns the values passed as arguments to the object on which setdata is called**

**void getdata(); //takes real and imag as input from user**

**float getreal(); //returns data member real**

**float getimaginary(); //returns data member imag**

**void display(); //displays the complex number in the form a+bi**

**};**#include<iostream>

using namespace std;

class complex

{

private:

float real; // Real Part

float imag; // Imaginary Part

public:

complex():real(0),imag(0)//Constructor

{}

complex(float r1,float i1)//2-arg constructor with default parameters

{

real=r1;

imag=i1;

}

complex add(complex a1)//Addition Function

{

real=real+a1.getreal(); // logic ofAddition of two complex numbers: (a + bi) + (c + di) = (a + c) + (b + d)i

imag=imag+a1.getimaginary();

complex r1(real,imag);

return r1; //return r1

}

complex multiply(complex a2)//multiplication function

{

real=(real\*a2.getreal())-(imag\*a2.getimaginary());//logic of Multiplication of two complex numbers: (a + bi)(c + di) = (ac − bd) + (bc + ad)i

imag=(imag\*a2.getreal())+(real+a2.getreal());

complex r2(real,imag);

return r2; //return r2

}

complex subtract(complex a3)//subtract function

{

real=real-a3.getreal();

imag=imag-a3.getimaginary();// logic ofSubtraction of two complex numbers: (a + bi) − (c + di) = (a − c) + (b − d)i

complex r3(real,imag);

return r3; //return r3

}

void divide()

{

//No logic....!

}

complex getconjugate()//function for conjucate

{

complex a4(real,imag);//Conjugate of a complex number: The conjugate of a + bi is a –bi

cout<<"Conjugate: "<<real<<-1\*imag<<"i"<<endl;

return a4;

};

void setdata(float re2,float im2 )//values passed as arguments to the object which are assing in setdata is called

{

real=re2;

imag=im2;

}

void getdata()//user input of real and imag.... part

{

cout<<"Enter Real part:";

cin>>real; //real part input

cout<<"Enter Imaginary Part:";

cin>>imag; //imag part input

}

float getreal()

{

return real;//returns data member real

}

float getimaginary()

{

return imag;//returns data member imag

}

void display() //output function

{

cout<<"Complex: "<<real<<"+"<<imag<<"i"<<endl; //displays the complex number in the form a+bi

}

}; //class end

int main()

{complex a[4];//class objects

a[0].getdata();//get from user

a[1].setdata(7,9);//set at addition

a[2].setdata(7,9);//set at multipilcation

a[3].setdata(7,9);//set at subtraction

cout<<"Addition of two complex numbers:"<<endl;//show complex numbers in addition

a[1].add(a[0]);

a[1].display();

a[0].getconjugate();

cout<<"Multiplication of two complex numbers:"<<endl;//show complex numbers in multiplycation

a[2].multiply(a[0]);

a[2].display();

a[0].getconjugate();

cout<<"Subtraction of two complex numbers:"<<endl;//show complex numbers in subtraction

a[3].multiply(a[0]);

a[3].display();

a[0].getconjugate();

system("pause");

return 0; //return 0 to main

}

**2**. Modify the following class to write one constructor equivalent to the three constructors so that the output of a program using the Point class remains the same.

**class Point**

**{**

**int x, y;**

**public:**

**Point():x(0),y(0){}**

**Point (int xVal):x(xVal),y(0){}**

**Point (int xVal,yVal):x(xVal),y(yVal) {}**

**void display( );**

**};**  
#include<iostream>

using namespace std;

class point//a class called point

{private:

int x,y;//two int variable

public:

//point():x(0),y(0){}

//point(int a):x(a),y(0){}

point(int a,int b):x(a),y(b){}//constructor

void dispaly() const//const function

{cout<<x<<y;}

};

int main()

{const point j(0,0);//object j with value

j.dispaly();//called funtion display

system("pause");

return 0;}

**3**. Create a class “Employee” that contains two data members: an employee number (of type int) and an employee compensation (of type float). Write a constructor that allows creation of objects with no, all or limited information. Member function getEmployee() should allow the user to enter employee’s data and a member showRecord() to display the information of the Employee. There should also be a member function setEmployee that allows changes the employee number or his compensation. Create atleast three employees and display their record.  
#include<iostream>

using namespace std;

class Employee//class employee

{

private:

int employee\_no;//int datatype employ number

float employee\_compensation;//float employee compensation

public:

void getEmployee1()//data of emplyee 1

{cout<<"Enter 1st Employee No:";

cin>>employee\_no;

cout<<"Enter 1st Employee Compensation:";

cin>>employee\_compensation;

}

void getEmployee2()//data of emplyee 2

{cout<<"Enter 2nd Employee No:";

cin>>employee\_no;

cout<<"Enter 2nd Employee Compensation:";

cin>>employee\_compensation;

}

void getEmployee3()//data of emplyee 3

{cout<<"Enter 3rd Employee No:";

cin>>employee\_no;

cout<<"Enter 3rd Employee Compensation:";

cin>>employee\_compensation;

}

void showRecord()//function for show data of employee

{cout<<"Employee No:"<<employee\_no<<endl;

cout<<"Employee Compensation:"<<employee\_compensation<<endl;

}

int setEmployee() //function foe change data

{int n;

cout<<"Enter 1 for change 1st Employee data,2 for 2nd Employee data,3 for 3rd Employee data:";

cin>>n;

switch(n)//switch statment

{case 1:

cout<<"Change 1st Employee No:";

cin>>employee\_no;

cout<<"Change 1st Employee Compensation:";

cin>>employee\_compensation;

break;

case 2:

cout<<"Enter 2nd Employee No:";

cin>>employee\_no;

cout<<"Enter 2nd Employee Compensation:";

cin>>employee\_compensation;

break;

case 3:

cout<<"Enter 3rd Employee No:";

cin>>employee\_no;

cout<<"Enter 3rd Employee Compensation:";

cin>>employee\_compensation;

default:

return 0;

break;

}}

};//end of class

int main()

{char c;

Employee c1,c2,c3;//creating object

c1.getEmployee1();//object 1

c2.getEmployee2();//object 2

c3.getEmployee3();//object 1

label://label of goto

c1.showRecord();//show information

c2.showRecord();//show information

c3.showRecord();//show information

cout<<"Press c for change data or any key to countinue:";

cin>>c;

if(c=='c')//condition for change

{c1.setEmployee();

goto label;//goto statment

}

system("pause");

return 0;}

**4**.Create a “Date” class that should consist of data members: month, day and year (all of type integers). The class should also have two member functions get\_date that allows the user to enter date in 31/12/97 (DD/MM/YY) format. Make a function show\_date that displays the date entered by user in the form “31 December 1997”.  
  
  
#include<iostream>

using namespace std;

class date//class date

{private:

int day;//day

int month;//month

int year;//year

public:

void get\_date()//get date from user

{cout<<"Enter date(day/month/year):";

cin>>day;

cout<<"/";

cin>>month;

cout<<"/";

cin>>year;}

void show\_date()//show date in alpha minths

{

switch(month)//switch statement for change numeric months to alpha

{

case 1:

cout<<day<<" January "<<19<<year<<endl;

break;

case 2:

cout<<day<<" Febuary "<<19<<year<<endl;

break;

case 3:

cout<<day<<" March "<<19<<year<<endl;

break;

case 4:

cout<<day<<" April "<<19<<year<<endl;

break;

case 5:

cout<<day<<" May "<<19<<year<<endl;

break;

case 6:

cout<<day<<" June "<<19<<year<<endl;

break;

case 7:

cout<<day<<" July "<<19<<year<<endl;

break;

case 8:

cout<<day<<" August "<<19<<year<<endl;

break;

case 9:

cout<<day<<" September "<<19<<year<<endl;

break;

case 10:

cout<<day<<" October "<<19<<year<<endl;

break;

case 11:

cout<<day<<" November "<<19<<year<<endl;

break;

case 12:

cout<<day<<" Decmber "<<19<<year<<endl;

break;

default:

cout<<"''Invalid Month''"<<endl;

}}

};//class end

int main()

{date tell;//object tell for access class funtions

tell.get\_date();//access get function for get date

tell.show\_date();//access show function for show date

system("pause");

return 0;}

**5.**Make a class Student with the data members String Name, int ID\_No, double GPA, int semesters. All data members should be kept private. The system should be able to create an instance of student even when the user has NO or limited record information of a student. There should be ONE constructor only. The class should also have functdisplaying the Students altering altering the number of semesters.  
  
  
  
#include<iostream>

#include<string.h>

using namespace std;

class Student//class student

{private:

char Name[50];

int ID\_No;

double GPA;

int semesters;

public:

Student():ID\_No(0)//constructor

{}

void get()//input function

{cout<<"Enter Student Name:";

cin.get(Name,50);

cout<<"Enter the ID Number:";

cin>>ID\_No;

cout<<"Enter the GPA:";

cin>>GPA;

cout<<"Enter Semester Number:";

cin>>semesters;

}

void Show()//display function

{strupr(Name);//for uppercase of name

cout<<"\nStudent ID# "<<ID\_No<<":"<<Name<<". His Grade Point Average till "<<semesters<<" is "<<GPA<<"."<<endl;}

}; //end class student

int main()

{Student re;//object re of student class

re.get();//get input from user

re.Show();//show output

system("pause");

return 0;

}

**THE END**