Term work

of

OOPs Using C++ Lab (PCS-307)

Submitted in partial fulfillment of the requirement for the III semester

Bachelor of Technology

By

PRIYANSHU KUMAR

2261445

Under the Guidance of

Mr. Ravindra Koranga

Assistant Professor

Deptt. of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GRAPHIC ERA HILL UNIVERSITY, BHIMTAL CAMPUS

SATTAL ROAD, P.O. BHOWALI

DISTRICT- NAINITAL-263132

2023-2024

Graphic Era
HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

CERTIFICATE

The term work of OOPs with C++ Lab (PCS-307), being submitted by Priyanshu Kumar s/o Puran Chadra, University Roll Number 2261445 to Graphic Era Hill University, Bhimtal Campus for the award of bona fide work carried out by him. He has worked under my guidance and supervision and fulfilled the requirement for the submission of this work report.

(..…………………)	**(……………………)**
Subject Professor	**HOD, CSE Dept.**

Graphic Era HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

ACKNOWLEDGEMENT

I take immense pleasure in thanking Honorable **Mr. Ravindra Koranga** (Assistant Professor, CS, GEHU Bhimtal Campus) for allowing us to carry out this project work under his excellent and optimistic supervision. This has all been possible due to his novel inspiration, able guidance and useful suggestions that have helped me in developing my subject concepts as a student.

I want to extend thanks to our President **"Prof. (Dr.) Kamal Ghanshala"** for providing us all infrastructure and facilities to work in need without which this work would not be possible.

(PRIYANSHU KUMAR)

pk9927652@gmail.com

Graphic Era
HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

STUDENT'S DECLARATION

I, Priyanshu Kumar, hereby declare the work, which is being presented in the report, entitled **Term work of OOPs with C++ Lab (PCS-307)** in partial fulfillment of the requirement for the award of the degree **Bachelor of Technology (Computer Science)** in the session **2023-2024** for semester III, is an authentic record of my own work carried out under the supervision of **Mr. Ravindra Koranga** (Graphic Era Hill University, Bhimtal)

The matter embodied in this project has not been submitted by me for the award of any other degree.

Date: …………   ……………….

(Full signature of student)

Graphic Era HILL UNIVERSITY
University under section 2(f) of UGC Act, 1956
Bhimtal Campus

Computer Science and Engineering Department
OOPS LAB (PCS-307)

Requirements:

- Unix/Linux based Computer System

Index/List of Experiments

1	An electricity board charges the following rates to domestic users to discourage large consumption of energy. For the first 100 units: - 60 P per unit. For the next 200 units: -80 P per unit Beyond 300 units: -90 P per unit All users are charged a minimum of Rs 50 if the total amount is more than Rs 300 then an additional surcharge of 15% is added. Implement a C++ program to read the names of users and number of units consumed and display the charges with names.
2	Construct a C++ program that removes a specific character from a given string and return the updated string. Typical Input: computer science is the future Typical Output: compuer science is he fuure
3	Implement a C++ program to find the non-repeating characters in string. Typical Input: graphic era university Typical Output: c g h n p s t u v y
4	Implement a C++ program to demonstrate the concept of data abstraction using the concept of Class and Objects.
5	Define a class Hotel in C++ with the following specifications Private members • Rno Data member to store room number • Name Data member to store customer name • Tariff Data member to store per day charges • NOD Data member to store number of days of stay • CALC() Function to calculate and return amount as NOD*Tariff ,and if the value of days* Tariff >10000,then total amount is 1.05* days*Tariff. Public members • Checkin() Function to enter the content Rno, Name, Tariff and NOD • Checkout() Function to display Rno, Name, Tariff, NOD and Amount (amount to be displayed by calling function) CALC()
6	Implement a Program in C++ by defining a class to represent a bank account.Include the following: Data Members ● Name of the depositor ● Account number ● Type of account (Saving, Current etc.) ● Balance amount in the account Member Functions ● To assign initial values ● To deposit an amount ● To withdraw an amount after checking the balance ● To display name and balance

7	Anna is a contender for valedictorian of her high school. She wants to know how many students (if any) have scored higher than her in the exams given during this semester. Create a class named Student with the following specifications: ➢ An instance variable named scores holds a student's 5 exam scores. ➢ A void input () function reads 5 integers and saves them to scores. ➢ An int calculateTotalScore() function that returns the sum of the student's Scores. Input Format: In the void Student::input() function, you must read 5 scores from standard input and save them to your scores instance variable. Output Format: In the int Student::calculateTotalScore() function, you must return the student's total grade (the sum of the values in scores). The code in the editor will determine how many scores are larger than Anna's and print that number to the console. Sample Input: The first line contains n, the number of students in Anna's class. The n subsequent lines contain each student's 5 exam grades for this semester. 3 30 40 45 10 10 40 40 40 10 10 50 20 30 10 10
8	Construct a Program in C++ to show the working of function overloading(compile Time polymorphism) by using a function named calculate Area () to calculate area Of square, rectangle and triangle using different signatures as required.
9	Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information As instance. Data Members - • partNumber (type String) • partDescription (type String) • quantity of the item being purchased (type int) • price_per_item (type double) Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a Method named getInvoiceAmount() that calculates the invoice amount (i.e., Multiplies the quantity by the price per item), then returns the amount as a double Value. If the quantity Is not positive, it should be set to 0. If the price per item Is not Positive,it should be set to 0.0. Write a test application named invoiceTest that Demonstrates class Invoice's capabilities.
10	Imagine a tollbooth with a class called TollBooth. The two data items are of type unsigned int and double to hold the total number of cars and total amount of money collected. A constructor initializes both of these data members to 0. A member function called payingCar( )increments the car total and adds 0.5 to the cash total. Another function called nonPayCar( ) increments the car total but adds nothing to the cash total. Finally a member function called display( )shows the two totals. Includee a program to test this class. This program should allow the user to push one key to count a paying car and another to count a non paying car. Pushing the ESC key should cause the program to print out the total number of cars and total cash and then exit.
11	Create a class called Time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it in 11:59:59 format. A member function named add() should add two objects of type time passed as arguments. A main ( ) program should create two initialized values together, leaving the result in the third time variable. Finally it should display the value of this third variable.
12	Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance

	variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest() to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by this interest should be added tosavingsBalance. Provide a static method modifyInterestRate() that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs2000.00 and Rs3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.
13	Create a class Complex having two int type variable named real & img denoting real and imaginary part respectively of a complex number. Overload +, - , == operator to add, to subtract and to compare two complex numbers being denoted by the two complex type objects.
14	Using the concept of operator overloading. Implement a program to overload the following: a. Unary – b. Unary ++ preincrement, postincrement c. Unary – predecrement, postdecrement
15	Using the concept of operator overloading. Implement a program to overload the following: With the help of friend function a. Unary – b. Unary ++ preincrement, postincrement c. Unary – predecrement, postdecrement

Q 1. An electricity board charges the following rates to domestic users to discourage Large consumption of energy. For the first 100 units: - 60 P per unit. For the next 200 units: -80 P per unit. Beyond 300 units: -90 P per unit. All users are charged a minimum of Rs 50 if the total amount is more than Rs 300. Then an additional surcharge of 15% is added. Implement a C++ program to read the names of users and number of units consumed and display the charges with names.

Code for the problem -

```
#include<iostream>
Using namespace std;
Int main()
{
    Int n;
    Float charge, scharge;
    String name;
    Cout<<" ¥n enter name and number of units consumed" ;
    Cin>>name;
    Cin>>n;
    If(n<=100)
        Charge=(0.60*n);
    If(n>100 && n<=300)
    {
        Charge=  100*.6 +(n-100)*0.80;
    }
    If(n>=300)
    {
        Charge= 100*.6 + 200*.8 +(n-300)*0.90;
    }
    If(charge<50)
        Charge=50;
    If(charge>300)
    {
        scharge=(0.15*charge);
        Charge= charge + .15*charge;
    }
Cout<<" electricity bill ¥n" ;
Cout<<name<<"  : : rs" <<charge;}
```

OUTPUT -

```
Enter name and number of units consumed Ram 400

Ram 400
Electricity bill
Ram : : rs356.
```

Q 2. Construct a C++ program that removes a specific character from a given string and Return the updated string.
Typical Input: computer science is the future
Typical Output: compuer science is he fuure

Code for the problem -
```
#include<iostream>
#include<string>
Using namespace std;
Int main()
{
    String s;
    String res=” ” ;
    Char c;
    Cout<<” input string  “<<endl;
    getline(cin,s);
    Cout<<” enter character to remove” <<endl;
    Cin>>c;
    For(int i=0;i<s.size();i++)
    {
        if(s[i]!=c)
        {
            res=res+s[i];
        }
    }
    Cout<<” The result string is :  “<<res<<endl;
}
```

OUTPUT -
```
Input string
Btech computer science
Enter character to remove
C


The result string is : Bteh omputer siene
```

Q 3. Implement a C++ program to find the non-repeating characters in string.

Typical Input: graphic era university

Typical Output: c g h n p s t u v y

Code for the problem -

```
#include<iostream>
#include<string>
Using namespace std;
Int main()
{
    String s;
    Cout<<" enter string ¥n" ;
    getline(cin,s);
    For(int i=0;i<s.size();i++)
    {
        Int duplicate=0;
        For(int j=0;j<s.size();j++)
        {
            if( s[i]==s[j] and i!=j)
            {
                Duplicate=1;
                break;
            }
        }
        If(duplicate==0)
        {
          Cout<<s[i]<<"  is unique character" <<endl;
        }
    }
}
```

OUTPUT -

```
Enter  String
Graphic era hill University -
G is unique character
P is unique character
C is unique character
U is unique character
N is unique character
V is unique character
S is unique character
T is unique character
Y is unique character
```

Q 4. Implement a C++ program to demonstrate the concept of data abstraction using the concept of Class and Objects.

Code for the problem -

```
#include<iostream>
Using namespace std;
#include<iostream>
Using namespace std;
Class student
{
   Private:
      String name;
      Int rn, eng, hin;
   Public:
      Void input();
      Void disp();
      Void total();
};
Void student::input()
{
    Cout<<" Enter name , rn, eng marks, hin marks" <<endl;
   Cin>>name>>rn>>eng>>hin;
}
Void student::disp()
{
    Cout<<" The details of students are  "<<endl;
   Cout<<name<<"   "<<rn<<"   "<<eng<<"   "<<hin<<endl;
}

Void student:: total()
{
    Cout<<" Total marks of students are" <<endl;
   Cout<<eng+hin<<endl; }
```

```
Int main()
{
   Student obj1, obj2;
   Obj1.input();
   Obj1.disp();
   Obj1.total();
   Obj2.input();
   Obj2.disp();
   Obj2.total();
}
```

OUTPUT -

```
Enter name , rn, eng marks, hin marks

Ram 73 45 36

The details of students are -

Ram 73 45 36

Total marks of students are -

81

Enter name , rn, eng marks, hin marks
```

Q 5. Define a class Hotel in C++ with the following specifications
Private members
- Rno Data member to store room number
- Name Data member to store customer name
- **Tariff Data member to store per day charges**
- NOD Data member to store number of days of stay
- **CALC() Function to calculate and return amount as NOD*Tariff ,and if the value of days* Tariff >10000,then total amount is 1.05* days*Tariff.**

Public members
- **Checkin() Function to enter the content Rno, Name, Tariff and NOD**
- **Checkout() Function to display Rno, Name, Tariff,**

NOD and Amount (amount to be displayed by calling function) CALC()

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class hotel
{
  Private:
        String name;
        Int room, nod, tariff;
        Float calc();
    Public:
        Void checkin();
        Void checkout();
};
Void hotel::checkin()
{
    Cout<<" Enter name, room#, no of days, tariff" <<endl;
    Cin>>name>>room>>nod>>tariff;
}
Void hotel::checkout()
{
    Cout<<" Details are as follows: ¥n" ;
    Cout<<name<<"   "<<room<<"   "<<nod<<"   "<<tariff<<endl;
    Cout<<" The total amount is  "<<calc()<<endl;;
}
Float hotel::calc()
{
  If(nod*tariff>10000)
    {
        Return nod*tariff*1.05;    }
    Else
    {
        Return nod*tariff;
    }
}
Int main()
{
    Int choice;
    Hotel obj;
    While(1)
    {
       Obj.checkin();
        Obj.checkout();
       Cout<<" press 0 to exit any other key to continue:" <<endl;
        Cin>>choice;
        If (choice==0)
            Break;
    }
}
```

OUTPUT -

```
Enter name, room#, no of days, tariff
Ram 26 8 250
Details are as follows:
Ram 26 8 250
The total amount is 2000
Press 0 to exit any other key to continue:
```

Q 6. Implement a Program in C++ by defining a class to represent a bank account.Include the following:

Data Members

- Name of the depositor
- Account number
- Type of account (Saving, Current etc.)
- Balance amount in the account

Member Functions

- To assign initial values
- To deposit an amount
- To withdraw an amount after checking the balance
- To display name and balance

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class bank
{
    Private:
        String name, type;
        Int account;
        Float balance;
```

```
     Public:
        Void input();
        Void disp();
        Void deposit(float amount);
        Void withdraw(float amount);
};
Void bank::input()
{
    Cout<<" Enter name , type, acc#, initial balance" <<endl;
    Cin>>name>>type>>account>>balance;
}

Void bank::disp()
{
    Cout<<" The details are ¥n" ;
    Cout<<name<<"   "<<type<<"   "<<account<<"   " <<balance<<endl;
}
Void bank::deposit(float amount)
{
    Balance= balance+amount;
    Cout<<" Amount deposited successfully¥n" ;
}
Void bank::withdraw(float amount)
{
    If(balance==0 or amount>balance)
    {
        Cout<<" Can' t Withdraw. Check Balance¥n" ;
    }
    Else
    {
        Balance=balance-amount;
    }
}
Int main()
{
    Bank obj;
    Cout<<" Object created. What operations do you want to perfornm¥n" ;
    Int choice;
    Float amount;
    While(1)
    {
        Cout<<" ¥n¥n 1. To input values¥n 2. To display values¥n 3. To
          Deposit balance¥n 4. To withdraw balance¥n 5. To Exit¥n" ;
Cin>>choice;
       Switch(choice)
        {
            Case 1:
                Obj.input();
                Break;
            Case 2:
                Obj.disp();
      Break;
           Case 3:
               Cout<<" enter amount to deposit¥n" ;
                Cin>>amount;
                Obj.deposit(amount);
                Break;
            Case 4:
                Cout<<" enter amount to withdraw¥n" ;
                Cin>>amount ;
                Obj.withdraw(amount);
                Break;
            Case 5:
                Return 0;
        }
    }
}
```

OUTPUT -

```
Object created. What operations do you want to perfornm
   1. To input values
   2. To display values
   3. To Deposit balance
   4. To withdraw balance
   5. To Exit
1
Enter name , type, acc#, initial balance
Ram save 626262 25678
   1. To input values
   2. To display values
   3. To Deposit balance
   4. To withdraw balance
   5. To Exit
3
Enter amount to deposit
```

```
3500
Amount deposited successfully
   1. To input values
   2. To display values
   3. To Deposit balance
   4. To withdraw balance
   5. To Exit
```

Q 7. Anna is a contender for valedictorian of her high school. She wants to know how many students (if any) have scored higher than her in the exams given during this semester.
Create a class named Student with the following specifications:
➢ An instance variable named scores holds a student's 5 exam scores.
➢ A void input () function reads 5 integers and saves them to scores.
➢ An int calculateTotalScore() function that returns the sum of the student's
Scores.
Input Format
In the void Student::input() function, you must read 5 scores from standard input and save them to your scores instance variable.
Output Format
In the int Student::calculateTotalScore() function, you must return the student's total grade (the sum of the values in scores).
The code in the editor will determine how many scores are larger than Anna's and print that number to the console.
Sample Input
The first line contains n, the number of students in Anna's class. The n subsequent lines contain each student's 5 exam grades for this semester.

```
3
30 40 45 10 10
40 40 40 10 10
50 20 30 10 10
```

Sample Output

```
1
```

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class student
{
    Private:
        String name;
        Int marks[5];
    Public:
        Void input();
        Void disp();
        Int total();
};
Void student::input()
{
    Cout<<" Enter name and marks of 5 subjects¥n" ;
    Cin>>name;
    For(int i=0;i<5;i++)
    {
        Cin>>marks[i];
    }
}
Void student::disp()
{
    Cout<<" Displaying name and marks in 5 subjects¥n" ;
    Cout<<name<<endl;
    For(int i=0;i<5;i++)
    {
        Cout<<marks[i] <<  "  ";    }
}
Int student::total()
{
    Int sum=0;
    For(int i=0;i<5;i++)
    {
        Sum+=marks[i];
    }
    Return sum;
}
Int main()
{
    Int n, val;
    Cout<<" Enter the marks of valedictorian¥n" ;
    Cin>>val;
    Cout<<” Enter the number of students¥n” ;
    Cin>>n;
    Student ar[n];
    For(int i=0;i<n;i++)
    {
        Ar[i].input();
    }
    Int count=0;
    For(int i=0;i<n;i++)
    {
        If(ar[i].total() > val)
        {
            Count++;
        }
    }
    Cout<<” The number of students having more marks is :  “<<count<<endl;
}
```

OUTPUT -

```
Enter the marks of valedictorian
265
Enter the number of students
3
Enter name and marks of 5 subjects
Ram 52 34 55 49 54
Enter name and marks of 5 subjects
Sham 56 48 51 39 51
Enter name and marks of 5 subjects
Sunny 54 43 39 54 58
The number of students having more marks is : 0
```

Q 8. Construct a Program in C++ to show the working of function overloading(compile Time polymorphism) by using a function named calculate Area () to calculate area Of square, rectangle and triangle using different signatures as required.

Code for the Problem -

```
#include<iostream>
#include<cmath>
Using namespace std;
Void area(int x)
{
    Cout<<" Area of square is  "<<x*x<<endl;
}
Void area(int x, int y)
{
    Cout<<" Area of rectangle is  "<<x*y<<endl;
}
Void area(int x, int y, int z)
{
    Float s=(float)(x+y+z)/2;
    Float area=sqrt(s*(s-x)*(s-y)*(s-z));
    Cout<<" Area=" <<area<<endl;
}
Int main()
{
    Int choice,x,y,z;
    While(1)
    {
        Cout<<" 1. For square¥n 2. For rectangle¥n 3. For triangle¥n 4. To
exit¥n” ;
        Cin>>choice;
        Switch(choice)
        {
            Case 1:
                Cout<<” enter side: ¥n” ;
                Cin>>x;
                Area(x);
                Break;
            Case 2:
                Cout<<” Enter len and breadth ¥n” ;
                Cin>>x>>y;
                Area(x,y);
                Break;
            Case 3:
                Cout<<” Enter s1, s2, s3 ¥n” ;
                Cin>>x>>y>>z;
                Area(x,y,z);
                Break;
            Case 4:
                Return 0;
        }
    }
}
```

OUTPUT -

```
   1. For square
   2. For rectangle
   3. For triangle
   4. To exit
1
Enter side:
4
Area of square is 16
   1. For square
   2. For rectangle
   3. For triangle
   4. To exit
2
Enter len and breadth
 4 8
Area of rectangle is 32
```

Q 9. Create a class called Invoice that a hardware store might use to represent an invoice for an item sold at the store. An Invoice should include four pieces of information As instance.
Data Members –

- partNumber (type String)
- partDescription (type String)
- quantity of the item being purchased (type int)
- price_per_item (type double)

Your class should have a constructor that initializes the four instance variables. Provide a set and a get method for each instance variable. In addition, provide a Method named getInvoiceAmount() that calculates the invoice amount (i.e., Multiplies the quantity by the price per item), then returns the amount as a double Value. If the quantity Is not positive, it should be set to 0. If the price per item Is not Positive,it should be set to 0.0. Write a test application named invoiceTest that Demonstrates class Invoice's capabilities.

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class invoice
{
    Int part_no, qty, price;
    String desc;
    Public:
    Invoice()
    {
       Part_no=0;
       Desc=” ” ;
       Qty=0;
       Price=0;
    }
    Void input()
    {
        Cout<<” Enter the values part no, description, qty, price ¥n” ;
        Cin>>part_no>>desc>>qty>>price;
        If(qty<0)
            Qty=0;
        If(price<0)
            Price=0;
    }
    Void disp()
    {
        Cout<<” Entered details are ¥n” ;
        Cout<<part_no<<”   “<<desc<<”   “<<qty<<”   “<<price<<endl;
 }
    Double getInvoiceAmount()
    {
        Double amount= qty*price;
        Return amount;
    }
};
Int main()
{
    Invoice obj;
    Int choice;
    While(1)
    {
        Cout<<” ¥n¥n****************************¥n” ;
        Cout<<” Enter choice 1. Input     2. Display
        3.Total amount       4. Exit¥n” ;
        Cin>>choice;
        Switch(choice)
        {
            Case 1:
                Obj.input();
                Break;
            Case 2:
                Obj.disp();
                Break;
            Case 3:
                Cout<<obj.getInvoiceAmount();
                Break;
            Case 4:
                Return 0;
   }
    }
   Return 0;
}
```

OUTPUT -

```
Enter choice 1. Input     2. Display    3. Total amount       4. Exit

1

Enter the values part no, description, qty, price

1234 Wheel 4 210

Enter choice 1. Input     2. Display    3. Total amount       4. Exit

3

840
```

Q 10. Imagine a tollbooth with a class called TollBooth. The two data items are of type unsigned int and double to hold the total number of cars and total amount of money collected. A constructor initializes both of these data members to 0. A member function called payingCar( )increments the car total and adds 0.5 to the cash total. Another function called nonPayCar( ) increments the car total but adds nothing to the cash total. Finally a member function called display( )shows the two totals. Includee a program to test this class. This program should allow the user to push one key to count a paying car and another to count a non paying car. Pushing the ESC key should cause the program to print out the total number of cars and total cash and then exit.

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class toll
{
    Private:
        Int count=0;
        Float total=0;
    Public:
        Void payingCar()
        {
            Cout<<" Paying car passed¥n" ;
            Count++;
            Total+=.5;
        }
        Void nonpayingCar()
        {
            Cout<<" Non Paying car passed¥n" ;
            Count++;
        }
        Void display()
        {
            Cout<<" The total count of cars is  "<<count<<endl;
            Cout<<" The total amount is  "<<total<<endl;
        }
};
Int main()
{
    Toll obj;

   Int choice;
    While(1)
    {
        Cout<<" ¥n¥n****************************¥n" ;
        Cout<<" Enter choice 1. Paying car     2. Non paying
        car  3. Exit¥n" ;
        Cin>>choice;
        Switch(choice)
        {
            Case 1:
                Obj.payingCar();
                Break;
            Case 2:
                Obj.nonpayingCar();
                Break;
            Case 3:
                Obj.display();
                Return 0;
        }
    }
    Return 0;
}
```

OUTPUT -

```
Enter choice 1. Paying car     2. Non paying car    3. Exit
1
Paying car passed
Enter choice 1. Paying car     2. Non paying car    3. Exit
2
Non Paying car passed
Enter choice 1. Paying car     2. Non paying car    3. Exit
3
The total count of cars is 2
The total amount is 0.5
```

Q 11. Create a class called Time that has separate int member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. A member function should display it in 11:59:59 format. A member function named add() should add two objects of type time passed as arguments. A main ( ) program should create two initialized values together, leaving the result in the third time variable. Finally it should display the value of this third variable.

Code for the Problem -

```
#include <iostream>
Using namespace std;
Class Time {
Private:
    Int hours;
    Int minutes;
    Int seconds;
Public:
    Void getTime(void);
    Void putTime(void);
    Void addTime(Time T1, Time T2);
};
Void Time::getTime(void)
{
    Cout << "Enter time:" << endl;
    Cout << "Hours? ";
    Cin >> hours;
    Cout << "Minutes? ";
    Cin >> minutes;
    Cout << "Seconds? ";
    Cin >> seconds;
}
Void Time::putTime(void)
{
    Cout << endl;
    Cout << "Time after add: ";
    Cout << hours << ":" << minutes << ":" << seconds << endl;
}
Void Time::addTime(Time T1, Time T2)
{
    This->seconds = T1.seconds + T2.seconds;
    This->minutes = T1.minutes + T2.minutes + this->seconds /
     60;

  This->hours = T1.hours + T2.hours + (this->minutes / 60);
    This->minutes %= 60;
    This->seconds %= 60;
}
Int main()
{
    Time T1, T2, T3;
    T1.getTime();
    T2.getTime();
    T3.addTime(T1, T2);
    T3.putTime();
    Return 0;
}
```

OUTPUT -

```
Enter time:
Hours? 9
Minutes? 45
Seconds? 12
Enter time:
```

```
Hours? 6
Minutes? 37
Seconds? 12
Time after add: 16:22:24
```

Q 12. Create class SavingsAccount. Use a static variable annualInterestRate to store the annual interest rate for all account holders. Each object of the class contains a private instance variable savingsBalance indicating the amount the saver currently has on deposit. Provide method calculateMonthlyInterest() to calculate the monthly interest by multiplying the savingsBalance by annualInterestRate divided by this interest should be added tosavingsBalance. Provide a static method modifyInterestRate() that sets the annualInterestRate to a new value. Write a program to test class SavingsAccount. Instantiate two savingsAccount objects, saver1 and saver2, with balances of Rs2000.00 and Rs3000.00, respectively. Set annualInterestRate to 4%, then calculate the monthly interest and print the new balances for both savers. Then set the annualInterestRate to 5%, calculate the next month's interest and print the new balances for both savers.

Code for the Problem -

```
#include<iostream>
```

```
Using namespace std;
Class SavingAccount
{private:
        Static double annualInterestRate;
        Double savingBalance;
    Public:
        Void inputBalance()
        {
            Cout<<" enter the value of balance:  ";
            Cin>>savingBalance;
        }
        Void monthlyInterest()
        {
            Double amount=(annualInterestRate*savingBalance)/12;
            Cout<<" the monthly interst interest is:
             "<<amount<<endl;
            savingBalance+=amount;
            cout<<" the balance is:  "<<savingBalance<<endl;
        }
        Void modifiedInterestRate(int x)
        {
            annualInterestRate=x;
        }
};
Double SavingAccount::annualInterestRate=4;
Int main()
{
    SavingAccount saver1,saver2;
    Saver1.inputBalance();
    Saver2.inputBalance();
    Saver1.monthlyInterest();
    Saver2.monthlyInterest();
    Cout<<" ---after modification---- "<<endl;
    Saver1.modifiedInterestRate(5);
    Saver1.monthlyInterest();
    Saver2.monthlyInterest();
    Return 0;
}
```

OUTPUT -

```
Enter the value of balance: 7000
Enter the value of balance: 8000
The monthly interst interest is: 2333.33
The balance is: 9333.33
The monthly interst interest is: 2666.67
The balance is: 10666.7
---after modification----
The monthly interst interest is: 3888.89
The balance is: 13222.2
The monthly interst interest is: 4444.44
The balance is: 15111.1
```

Q 13. Create a class Complex having two int type variable named real & img denoting real and imaginary part respectively of a complex number. Overload +, - , == operator to add, to subtract and to compare two complex numbers being denoted by the two complex type objects.

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class complex
{
    Private:
        Int real;
        Int img;
    Public:
        Complex(): real(0),img(0) { }
        Complex(int r,int i)
        {
            Real=r;
            Img =I;
        }
        Complex operator+(complex c)
        {
            Complex temp;
            Temp.real=real+c.real;
            Temp.img=img+c.img;
            Return temp;
        }
        Complex operator-(complex c)
        {
            Complex temp;
            Temp.real=real-c.real;
            Temp.img=img-c.img;
            Return temp;
        }
        Int operator==(complex c)
        {
            Complex temp;
           If(real==c.real && img==c.img)
                Return 1;
            Else
                Return 0;
        }
        Void display()
        {
            Cout<<" real part:  "<<real<<"  imaginary:
            "<<img<<endl;
        }
};
Int main()
{
    Complex obj1(5,4),obj2(4,5),obj3,obj4;
    Obj3=obj1+obj2;
    Obj1.display();
    Obj2.display();
    Obj3.display();
    Obj4=obj1-obj2;
    Obj4.display();
    If(obj1==obj2)
        Cout<<" both the objet are same" <<endl;
    Else
        Cout<<" both the objects are not same" <<endl;
    Return 0;
```

```
}
```

OUTPUT -

```
Real part: 5 imaginary: 4
Real part: 4 imaginary: 5
Real part: 9 imaginary: 9
Real part: 1 imaginary: -1
Both the objects are not same
```

Q 14. Using the concept of operator overloading. Implement a program to overload the following:

   a. Unary -
   b. Unary ++ preincrement, postincrement
   c. Unary - predecrement, postdecrement

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class test
{
    Private:
        Static int count;
        Int num;
```

```
Public:
    Test()
    {
        Count++;
    }
    Void input()
    {
        Cout<<” enter value of num:  “;
        Cin>>num;
    }
    Test operator-()
    {
        Test temp;
        Temp.num=-num;
        Return temp;
    }
    Test operator++(int)
    {
        Test temp;
        Temp.num=num++;
        Return temp;
    }
    Test operator++()
    {
        Test temp;
        Temp.num=++num;
        Return temp;
    }
    Test operator-(int)
    {
        Test temp;
        Temp.num=num--;
        Return temp;
    }
    Test operator-()
    {
        Test temp;
        Temp.num=--num;
        Return temp;
    }
    Void output()
    {
        Cout<<” the value of number:  “<<num<<endl;
    }
```

```
};
Int test::count=0;
Int main()
{
    Test obj1,min,preinc,postinc,predec,postdec;
    Obj1.input();
    Cout<<" ------minus-------- "<<endl;
    Min=-obj1;//obj1.operator-();
    Obj1.output();
    Min.output();
    Cout<<" ------pre increment-------- "<<endl;
    Preinc=++obj1;
    Obj1.output();
    Preinc.output();
    Cout<<" ------post increment-------- "<<endl;
    Postinc=obj1++;
    Obj1.output();
    Postinc.output();
    Cout<<" ------pre decrement-------- "<<endl;
    Predec=--obj1;
    Obj1.output();
    Predec.output();
    Cout<<" ------post decrement-------- "<<endl;
    Postdec=obj1--;
    Obj1.output();
    Postdec.output();
    Return 0;
}
```

OUTPUT -

```
Enter value of num: 9

-------minus---------
The value of number: 9
The value of number: -9
-------pre increment---------
The value of number: 10
The value of number: 10
------post increment--------
The value of number: 11
The value of number: 10
-------pre decrement---------
The value of number: 10
The value of number: 10
------post decrement--------
The value of number: 9
The value of number: 10
```

Q 15. Using the concept of operator overloading. Implement a program to overload the following:
With the help of friend function

a. **Unary -**
b. **Unary ++ preincrement, postincrement**
c. **Unary - predecrement, postdecrement**

Code for the Problem -

```
#include<iostream>
Using namespace std;
Class test
{
    Private:
        Static int count;
        Int num;
    Public:
        Test()
        {
            Count++;
        }
        Void input()
        {
            Cout<<" enter value of num:  ";
            Cin>>num;
        }
        Friend test operator-(test& obj)
        {
            Test temp;
            Temp.num=-obj.num;
            Return temp;
        }
        Friend test operator++(test& obj,int)
        {
            Test temp;
            Temp.num=obj.num++;
            Return temp;
        }
        Friend test operator++(test& obj)
        {
            Test temp;
            Temp.num=++obj.num;
            Return temp;
        }
        Friend test operator-(test& obj,int)

        {
            Test temp;
            Temp.num=obj.num--;
            Return temp;
        }
        Test operator-()
        {
            Test temp;
            Temp.num=--num;
            Return temp;
        }
        Void output()
        {
            Cout<<" the value of number:  "<<num<<endl;
        }
};
Int test::count=0;
Int main()
{
    Test obj1,min,preinc,postinc,predec,postdec;
    Obj1.input();
    Cout<<" ------minus-------- "<<endl;
    Min=-obj1;//obj1.operator-();
    Obj1.output();
    Min.output();
    Cout<<" ------pre increment-------- "<<endl;
    Preinc=++obj1;
    Obj1.output();
    Preinc.output();
    Cout<<" ------post increment-------- "<<endl;
    Postinc=obj1++;
    Obj1.output();
    Postinc.output();
    Cout<<" ------pre decrement-------- "<<endl;
    Predec=--obj1;
    Obj1.output();
    Predec.output();
    Cout<<" ------post decrement-------- "<<endl;

    Postdec=obj1--;

    Obj1.output();

    Postdec.output();

    Return 0;

}
```

OUTPUT -

```
Enter value of num: 6
-------minus---------

The value of number: 6

The value of number: -6

-------pre increment---------

The value of number: 7

The value of number: 7

------post increment--------

The value of number: 8

The value of number: 7

-------pre decrement---------

The value of number: 7

The value of number: 7

------post decrement--------

The value of number: 6

The value of number: 7
```