Course: OOPS **Session:** Operator Overloading **Timestamp:** 2021-4-1 10:24:09 **Register Number:** RA2031241010094

Q. Travel

When we travel positive distance means travelling forward and negative means travelling backwards.
Your task is to overload the unary + and unary - operator to display the same.
Mandatory:
1.Create a class Distance(feet,inches)
2.Overload operator - to calculate distance traveled backwards
3.Overload operator + to calculate distance traveled forward
4.Create a method named "displayDistance" to display the traveled distance.
Input :
First Line contains Distance(Feet and Inches separated by space)
Refer Sample test cases.
Programming language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class Distance
{
  int feet;
  int inches;
  public:
  Distance()
  {
    feet=0;
    inches=0;
  }
  Distance(int f,int i)
  {
    feet=f;
    inches=i;
  }
  void displayDistance()
  {
    cout<<"Feet="<<feet<<" Inches="<<inches<<endl;
  }
  Distance operator-()
  {
    cout<<"Travelling Backwards"<<endl;
    return Distance(feet,inches);
  }
  Distance operator+()
  {
    cout<<"Travelling Forward"<<endl;
    return Distance(feet,inches);
  }
};
int main()
{
  int a,b;
  cin>>a>>b;
  Distance D1(a,b);
  +D1;
  D1.displayDistance();
  -D1;
  D1.displayDistance();
 return 0;
}
```

Sample Input

```
10
19
```

Sample Output

```
Travelling Forward
Feet=10 Inches=19
Travelling Backwards
Feet=10 Inches=19
```

Result

Thus, Program " **Travel** " has been successfully executed

Q. Decimal Decrement

Your task is to overload the prefix decrement operator ++ to decrement the digit after decimal.
Mandatory:
1. Create a class named as "Decimal"
2. Declare the public data member and define the member variable.
3. Use the function named as "operator --()" of void type to increase the decimal value.
4. Create an object named "obj" for the Decimal class.
5. Access the function "operator --()" using the object of Decimal class and print the result in main method.

Refer Sample test cases.
Programming language need to be used:C+

Source Code

```
#include <iostream>
using namespace std;
class Decimal
{
  float a;
  public:
  void in()
  {
    cin>>a;
  }
  void operator --()
  {
    a=a-0.1;
    cout<<a;
  }
};
int main()
{
  Decimal obj;
  obj.in();
  obj.operator --();
  return 0;
}
```

Sample Input

```
17.8
```

Sample Output

```
17.7
```

Result

Thus, Program " **Decimal Decrement** " has been successfully executed

Q. Light House

Light House in charge in maria beach is interested in sending the current time to his official in proper time format when he joins the duty in the morning and when he leave the duty in the night.

Can you help him to convert the input time into proper format and to display it?

Mandatory:

1.Create a class Time with data members hours,mins,secs.

2.Overload operator <

3.Overload operator >> to get the input time

Input Format :

First Line contains Time (hours mins secs)

Refer Sample test cases.

Programming language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class Time
{
  int hours,mins,secs;
  public:
  void operator >>(int b)
  {
    cin>>hours>>mins>>secs;
  }
  void operator <<(int b)
  {
    cout<<hours<<" Hours "<<mins<<" Mins "<<secs<<" secs";
  }
};
int main()
{
  Time t;
  int a=3;
  t.operator >>(a);
  t.operator <<(a);
 return 0;
}
```

Sample Input

```
20 10 12
```

Sample Output

```
20 Hours 10 Mins 12 secs
```

Result

Thus, Program " **Light House** " has been successfully executed

Course: OOPS **Session:** Operator Overloading **Timestamp:** 2021-4-1 10:25:48 **Register Number:** RA2031241010094

Q. Savings

A Savings calculator can help you understand how long it will take to save a specific amount, or how much you need to save to have enough by a particular date.

Total Savings is calculated by subtracting expenditure from salary and then adding it to the initial savings.

Mandatory:

1.Create a class Money which takes both rupees and Paise as members of class.

2. Calculate the savings by overloading + operator and - operator for the class Money.

Money operator +(Money o)

Money operator -(Money o)

Input Format:

First Line contains the initial Savings(both Rupees and paise separated by space)

Second Line contains the Salary in a month (both Rupees and paise separated by space)

Third Line contains the expenditure in a month (both Rupees and paise separated by space)

Refer Sample testcases.

Programming language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class Money
{
  private:
  int rupees,Paise;
  public:
  Money()
  {
    cin>>rupees>>Paise;
  }
  Money operator +(Money o)
  {
    Money temp;
    temp.rupees=rupees+o.rupees;
    temp.Paise=Paise+o.Paise;
    return temp;
  }
  Money operator -(Money o)
  {
    Money temp;
    temp.rupees=rupees-o.rupees;
    temp.Paise=Paise-o.Paise;
    return temp;
  }
  void display()
  {
    cout<<"Rs="<<rupees<<" and "<<Paise<<" Paise"<<endl;
  }
};
int main()
{
  Money M1,M2,M3,M4,M5;
  M4=M2-M3;
  M5=M1+M4;
  M5.display();
  return 0;
}
```

Sample Input

```
10000 25
5000 75
2000 25
```

Sample Output

```
Rs=13000 and 75 Paise
```

Result

Thus, Program " **Savings** " has been successfully executed

Q. First Day of College

On the first day of the college ,three students named P,Q,R who were strangers wanted to know each other's addresses .

Being mathematical students,P and Q said their house addresses in the form of vector numbers which represents directions, of the form (ai+bj+cz) house can be obtained by adding the directions of P and Q.help them in finding the directions of R using operator overloading;

Hints:
1. Create class named "vector"
2. Define a class with 3 vars namely x,y,z;
3. Read the 3 directions and finally print the address of R.
4. overload + operator (as vector operator+(vector b)()) inside the same class and return the result.

Refer sample test cases.

Programming language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class vector
{
  private:
  int x,y,z;
  public:
  vector()
  {
    cin>>x>>y>>z;
  }
  vector operator+(vector b)
  {
    vector temp;
    temp.x=x+b.x;
    temp.y=y+b.y;
    temp.z=z+b.z;
    cout<<"Sum="<<temp.x<<"i+"<<temp.y<<"j+"<<temp.z<<"z"<<endl;
  }
};
int main()
{
  vector a,b,c;
  c=a+b;
 return 0;
}
```

Sample Input

```
2 4 6
1 4 5
```

Sample Output

```
Sum=3i+8j+11z
```

Result

Thus, Program " **First Day of College** " has been successfully executed

Course: OOPS **Session:** Operator Overloading **Timestamp:** 2021-4-1 10:26:40 **Register Number:** RA2031241010094

Q. Concatenate

Your task is to Concatenate two given strings using Overloading + operator.
Mandatory:
1. Create the class name as "concatenate".
2. Declare public data member and define the variable.
3. Using the function read() to get the input string.
4. Define the functions "operator +" and access the looping to concatenate the strings.
5. Create an object named "obj" for the concatenate class.
6. Access the function read() using the object of concatenate class and print the result in main method.

Refer Sample Test Cases.
Programming Language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class concatenate
{
  char a[100],b[100];
  public:
  void read()
  {
    cin>>a>>b;
  }
  void operator +()
  {
    cout<<a<<b;
  }
};
int main()
{
  concatenate obj;
  obj.read();
  obj.operator+();
 return 0;
}
```

Sample Input

```
Happy
Programming
```

Sample Output

```
HappyProgramming
```

Result

Thus, Program " **Concatenate** " has been successfully executed

Q. Decimal Increment

Your task is to overload the prefix increment operator ++ to increment the digit after decimal.
Mandatory:
1. Create a class named as "Decimal"
2. Declare the public data member and define the member variable.
3. Use the function named as "operator ++()" of void type to increase the decimal value.
4. Create an object named "obj" in main for the Decimal class.
5. Access the function "operator ++()" using the object of Decimal class and print the result in main method.

Refer Sample test cases.
Programming language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class Decimal
{
  float a;
  public:
  void in()
  {
    cin>>a;
  }
  void operator ++()
  {
    a=a+0.10;
    cout<<a;
  }
};
int main()
{
  Decimal obj;
  obj.in();
  obj.operator ++();
  return 0;
}
```

Sample Input

```
12.7
```

Sample Output

```
12.8
```

Result

Thus, Program " **Decimal Increment** " has been successfully executed

Q. Play with Fraction

Your task is to perform addition of fraction(normalization is not required) by overloading the + operator.

Create a class Fraction with two variables numerator and denominator.

Input Method:

Line 1: First line consists of the first fraction with numerator and denominator separated by space.
Line 2: Second line consists of the second fraction with numerator and denominator separated by space.

Mandatory:

1. Create a class named as "Fraction".
2. Declare the public data member and define member variable.
3. Using the "operator+" of Fraction class to perform the addition of fraction.
4. Create an object named "obj" for the Fraction class.
5. Access the operator of Fraction class and print the result in main method.

Refer Sample testcases.

Programming languages need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class Fraction
{
 public:
 int num,den;
 Fraction()
 {
  num=0;
  den=0;
 }
 void getinput()
 {
  cin>>num>>den;
 }
 Fraction operator+(Fraction obj)
 {
  Fraction temp;
  temp.num=(num*obj.den)+(den*obj.num);
  temp.den=den*obj.den;
  return temp;
 }
};
int main()
{
 Fraction f1,f2,add;
 f1.getinput();
 f2.getinput();
 add=f1+f2;
 cout<<add.num<<"/"<<add.den;
return 0;
}
```

Sample Input

```
6 3
8 4
```

Sample Output

```
48/12
```

Result

Thus, Program " **Play with Fraction** " has been successfully executed

Q. Ice Cream Seller

An ice-cream stall sells both green tea and mocha ice cream.

A small portion of either costs $0.75 and a large portion costs $1.25.

During a short period of time, the number of ice creams sold is taken as a matrix(2*2) which contains the no of small and large portions of both flavours.

Find out the sales of the green tea and mocha flavour.

Mandatory:
1. class name is matrix
2. Overload the operator * as follows
matrix operator *
3. Create a method named "get" of type void to get the inputs.
4. Create a method named "put" of type void to print the outputs.

Input Format:

Input:First and Second line contains the small and large portion of green tea.

Third and Fourth line contains the small and large portion of mocha.

Fifth and sixth line contains the price of small and large portion respectively.

Source Code

```
#include <iostream>
using namespace std;
class matrix
{
  matrix operator * ()
  {

  }
  void get()
  {
  }
  void put()
  {
  }
};
int main()
{
   double a[10][10],b[10][10],multi[10][10],r1,c1,r2,c2;
    int i,j,k;
    r1=2;c1=2;
    r2=2;c2=1;
    while(c1!=r2)
            {
              cin>>r1>>c1;
              cin>>r2>>c2;
            }
  for(i=0;i<r1;++i)
         for(j=0;j<c1;++j)
         {
           cin>>a[i][j];
         }
   for(i=0;i<r2;++i)
              for(j=0;j<c2;++j)
              {
                cin>>b[i][j];
              }
            for(i=0;i<r1;++i)
              for(j=0;j<c2;++j)
              {
                cin>>multi[i][j];
              }
            for(i=0;i<r1;++i)
              for(j=0;j<c2;++j)
                for(k=0;k<c1;++k)
                {
                  multi[i][j]+=a[i][k]*b[k][j];
                }
            for(i=0;i<r1;++i)
              for(j=0;j<c2;++j)
              {
                cout<<multi[i][j];
                if(j==c2-1)
                  cout<<endl;
              }
 return 0;
}
```

Sample Input

```
3
4
6
3
0.75
1.25
```

Sample Output

```
7.25
8.25
```

Result

Thus, Program " **Ice Cream Seller** " has been successfully executed

Q. Unary

Your task is to change the sign of a given data object by overloading unary operator.
Mandatory:
1. Create the class name as "data".
2. Declare public data member function and define the variable.
3. Using the function setdata() to get the two numbers.
4. Create an object named "obj" for the data class.
5. Access the function setdata() using the object of data class and print the result in main method.

Refer Sample Testcases.
Programming Language need to be used:C++

Source Code

```
#include <iostream>
using namespace std;
class data
{
  int a,b,c,d;
  public:
  data()
  {
        cin>>a>>b;
    cin>>c>>d;
  }
  void setdata()
  {
    cout<<(-1)*a<<" ";
    cout<<(-1)*b<<endl;
    cout<<(-1)*c<<" "<<(-1)*d;
  }
};
int main()
{
  data obj;
  obj.setdata();
 return 0;
}
```

Sample Input

```
10 20
8 20
```

Sample Output

```
-10 -20
-8 -20
```

Result

Thus, Program " **Unary** " has been successfully executed