Name:-Mizanul Hoque.
ID;-C221101.

Topic:-Operator overloading.

Remember:-
/*When a binary operator is overloaded, the left operand is passed implicitly to the function and the right operand is passed as an argument*/

BINARY OPERATOR OVERLOADING (+)

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
   int x,y;
public:
   coord()
   {
      x=0;
      y=0;
   }
   coord(int i,int j)
   {
      x=i;
      y=j;
   }
   void show()
   {
      cout<<x<<" "<<y<<endl;
   }
   coord operator + (coord ob1)
   {
      coord temp;
      temp.x=x+ob1.x;
      temp.y=y+ob1.y;
      return temp;
   }
};
int main()
{
   coord o1(1,2),o2(5,5),o3,o4(3,3);
    o3=o1+o2+o4;
    o3.show();
```

```
    return 0;
}
```

BINARY OPERATOR OVERLOADING (-)

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
    coord operator - (coord ob1)
    {
        coord temp;
        temp.x=x-ob1.x;
        temp.y=y-ob1.y;
        return temp;
    }
};

///DECLERATION OUTSIDE THE CLASS
//coord coord ::operator - (coord ob1){}
//return_type class_name :: operator symbol (argument)
int main()
{
    coord o1(1,2),o2(5,5),o3,o4(3,3);
     o3=o1-o2-o4;
     o3.show();
    return 0;
}
```

BINARY OPERATOR OVERLOADING (=)

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
   int x,y;
public:
   coord()
   {
      x=0;
      y=0;
   }
   coord(int i,int j)
   {
      x=i;
      y=j;
   }
   void show()
   {
      cout<<x<<" "<<y<<endl;
   }
   coord operator = (coord ob1)
   {
      x=ob1.x;
      y=ob1.y;
      return *this;///(Update compiler hole na likle o cole).
   }
};

int main()
{
   coord o1(1,2),o2(5,5),o3,o4(3,3);
    o3=o1;
    o3.show();
    o3=o2;
    o3.show();
    o3=o4;
    o4.show();
   return 0;
}
```

BINARY OPERATOR OVERLOADING (+)(When argument as an integer)

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
   int x,y;
public:
   coord()
   {
      x=0;
      y=0;
   }
   coord(int i,int j)
   {
      x=i;
      y=j;
   }
   void show()
   {
      cout<<x<<" "<<y<<endl;
   }
   coord operator + (int i)
   {
      coord temp;
      temp.x=x+i;
      temp.y=y+i;
      return temp;
   }
};

int main()
{
   coord o1(1,2),o2(5,5),o3,o4(3,3);
    o3=o4+1;
    ///o3=1+o4(!!Mind that, It is impossible)(Bacause when we pass an argument
   ///it automatically passed an class object in this overloaded function)
    o3.show();
   return 0;
}
```

LOGICAL OPERATOR OVERLOADING (==)

```
#include<bits/stdc++.h>
using namespace std;
```

```
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
    ///int operator ==(coord ob1)//(eibabe o lika jabe).
    bool operator ==(coord ob1)/// logical operator er function gula true or false return kore.
    {
        return x==ob1.x && y==ob1.y;
    }
};
int main()
{
    coord o1(1,2),o2(5,5),o3(5,5),o4(3,3);
     if(o3==o2)
     o2.show();
     else
        cout<<"WA"<<endl;
    return 0;
}
```

LOGICAL OPERATOR OVERLOADING (&&)

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
public:
    coord()
    {
```

```
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
    bool operator && (coord ob1)/// logical operator er function gula true or false return kore.
    {
        return (x && ob1.x) && (y && ob1.y);
    }
};
int main()
{
    coord o1(1,2),o2(5,5),o3(5,5),o4(3,0);
     if(o3 && o4)
     o2.show();
     else
        cout<<"WA"<<endl;
    return 0;
}
```

UNARY OPERATOR OVERLOADING (++)prefix increment

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
```

```
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
    coord operator ++()
    {
        x++;
        y++;
        ///return *this;//(Update compiler hole na likle o cole).
    }
};
int main()
{
    coord o1(10,20),o2(5,5),o3,o4(3,3);
     ++o4;
     ///o1++;///eirkm dile code bhul hobe.
     ///karon basically, eita prefix increment kore.
     ///jodi postfix increment korte cai argumnet er eikane
     ///(int not used )ikte hobe eita niyom;
     ///nicer code deko;
     o4.show();
    return 0;
}
```

UNARY OPERATOR OVERLOADING (++)postfix increment

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
```

```
    {
        cout<<x<<" "<<y<<endl;
    }
    coord operator ++(int notused)
    {
        x++;
        y++;
        return *this;
    }
};
int main()
{
    coord o1(10,20),o2(5,5),o3,o4(3,3);
     o1++;
     o1.show();
    return 0;
}
```

UNARY OPERATOR OVERLOADING (-)negation

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
    coord operator -()
    {
        x=-x;
        y=-y;
        return *this;///eiketre eita dite hobe.
        ///karon ami jokon eita carav run dici garbage value cole asce.
```

```
    }
};
int main()
{
    coord o1(10,20),o2(5,5),o3,o4(3,3);
     o1=-o1;
     o1.show();
    return 0;
}
```

USING FRIEND FUNCTION OPERATOR OVERLOADING (+)

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
    friend coord operator + (coord ob1,coord ob2);
};
coord operator + (coord ob1,coord ob2)
{
    coord temp;
    temp.x=ob1.x+ob2.x;
    temp.y=ob1.y+ob2.y;
    return temp;
}
int main()
{
    coord o1(1,2),o2(5,5),o3,o4(3,3);
    o3=o1+o2+o4;
```

```
    o3.show();
    return 0;
}
```

USING FRIEND FUNCTION OPERATOR OVERLOADING (+)(ANOTHER TYPE OBJECT PASS (1))

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
    friend coord operator + (coord ob1,int i);
};
coord operator + (coord ob1,int i)
{
    coord temp;
    temp.x=ob1.x+i;
    temp.y=ob1.y+i;
    return temp;
}
int main()
{
    coord o1(1,2),o2(5,5),o3,o4(3,3);
    o3=o1+1;
    ///o3=1+o1(It's Wrong).(MInd it)..
    o3.show();
    return 0;
}
```

USING FRIEND FUNCTION OPERATOR OVERLOADING (+)(ANOTHER TYPE OBJECT PASS(2))

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
   int x,y;
public:
   coord()
   {
      x=0;
      y=0;
   }
   coord(int i,int j)
   {
      x=i;
      y=j;
   }
   void show()
   {
      cout<<x<<" "<<y<<endl;
   }
   friend coord operator + (int i,coord ob1);
};
coord operator + (int i,coord ob1)
{
   coord temp;
   temp.x=i+ob1.x;
   temp.y=i+ob1.y;
   return temp;
}
int main()
{
   coord o1(1,2),o2(5,5),o3,o4(3,3);
   o3=2+o1;
   ///o3=o1+2(It's Wrong).(MInd it)..
   o3.show();
   return 0;
}
```

USING FRIEND FUNCTION UNARY OPERATOR OVERLOADING (++)prefix increment(Use reference parameter)

```
#include<bits/stdc++.h>
using namespace std;
```

```
class coord
{
    int x,y;
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
  friend coord operator ++(coord &ob1);

};
    coord operator ++(coord &ob1)
    {
        ob1.x++;
        ob1.y++;
        return ob1;
    }
int main()
{
    coord o1(10,20),o2(5,5),o3,o4(3,3);
     ++o4;
     o4.show();
    return 0;
}
```

USING FRIEND FUNCTION UNARY OPERATOR OVERLOADING (++)postfix increment(Use reference parameter)

```
#include<bits/stdc++.h>
using namespace std;
class coord
{
    int x,y;
```

```
public:
    coord()
    {
        x=0;
        y=0;
    }
    coord(int i,int j)
    {
        x=i;
        y=j;
    }
    void show()
    {
        cout<<x<<" "<<y<<endl;
    }
  friend coord operator ++(coord &ob1,int notused);

};
    coord operator ++(coord &ob1,int notused)
    {
        ob1.x++;
        ob1.y++;
        return ob1;
    }
int main()
{
    coord o1(10,20),o2(5,5),o3,o4(3,3);
     o1++;
     o1.show();
    return 0;
}
```