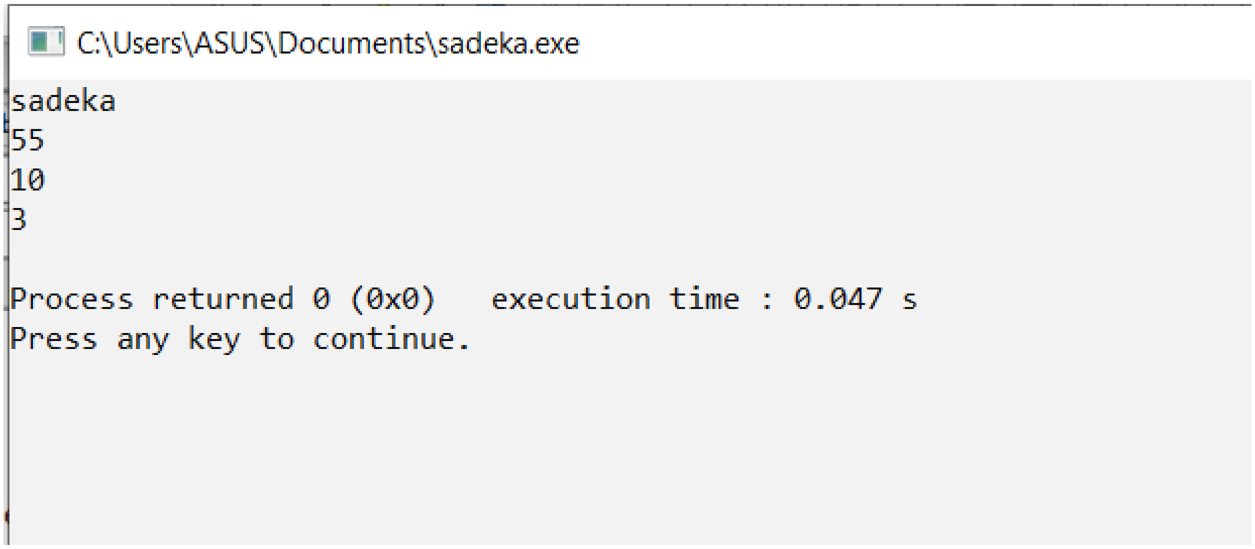
01.

```
#include<iostream>
using namespace std;
class university
{ public:
   int num_of_dept;
   int rank;
   university(int a,int b)
   {
      num_of_dept=a;
      rank=b;
   }
  virtual  void show()
   {
      cout<<num_of_dept<< endl;
      cout<<rank<<endl;
   }
};
 class department: public university
 { public:
    string name;
 int student_num;
 department(string x,int y,int m,int n):university(m,n)
 { name=x;
 student_num=y;
 }
 void show()
 {
    cout<<name<<endl;

    cout<<student_num<<endl;

    cout<<num_of_dept<<endl;

    cout<<rank<<endl;

 }

 };

int main()

{

   department ob("sadeka", 55,10,3);

   ob.show();

 return 0;


}
```

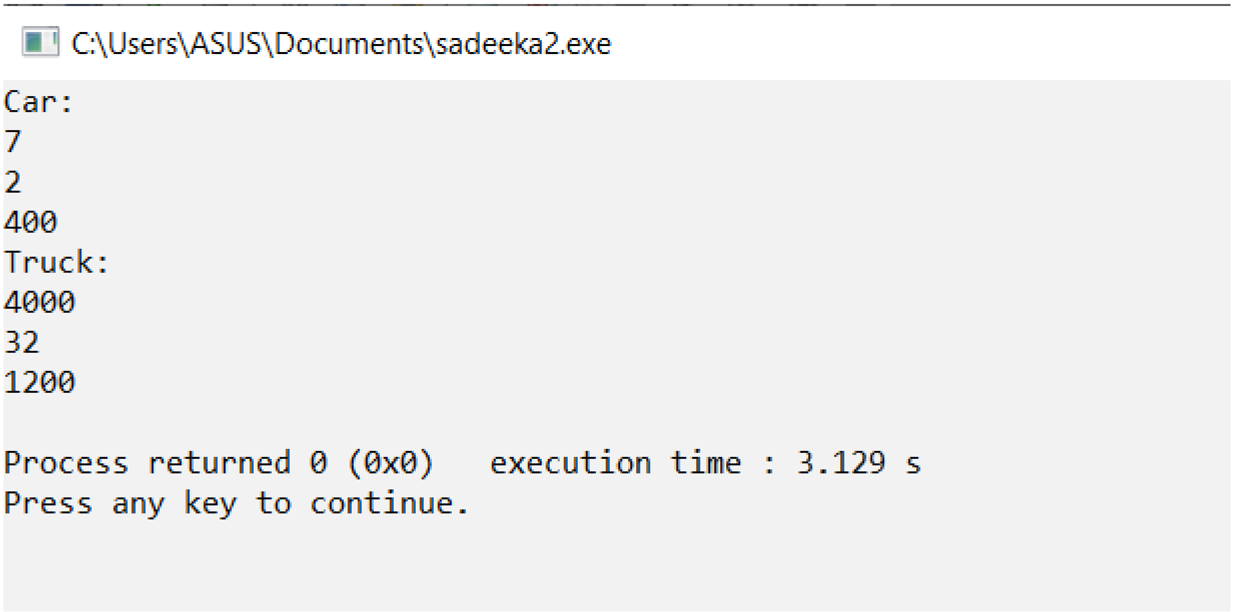


```
sadeka
55
10
3

Process returned 0 (0x0)   execution time : 0.047 s
Press any key to continue.
```

02.

```
#include<iostream>
using namespace std;
class vehicle
{ public:
  int wheels, range;
  vehicle(int a,int b)
  {
      wheels=a;
      range=b;
  }
  void show()
  {
      cout<<wheels<<endl;
      cout<<range<<endl;
  }
};
class car:public vehicle
{
public:
    int passengers;
    car(int x,int y,int z):vehicle(y,z)
    {
        passengers=x;
    }
    void show1()
    { cout<<"Car:"<<endl;
        cout<<passengers<<endl;
```

```
   cout<<wheels<<endl;
   cout<<range<<endl;
    }
};
class truck:public vehicle
{ public:
   int loadlimit;
   truck(int m,int n,int o):vehicle(n,o)
   {
      loadlimit=m;
   }
   void show2()
   {
       cout<<"Truck:"<<endl;
       cout<<loadlimit<<endl;
       cout<<wheels<<endl;
   cout<<range<<endl;;
   }
};
int main()
{
   car c(7,2,400);
   c.show1();
   truck t(4000,32,1200);
   t.show2();
   return 0;
}
```



C:\Users\ASUS\Documents\sadeeka2.exe

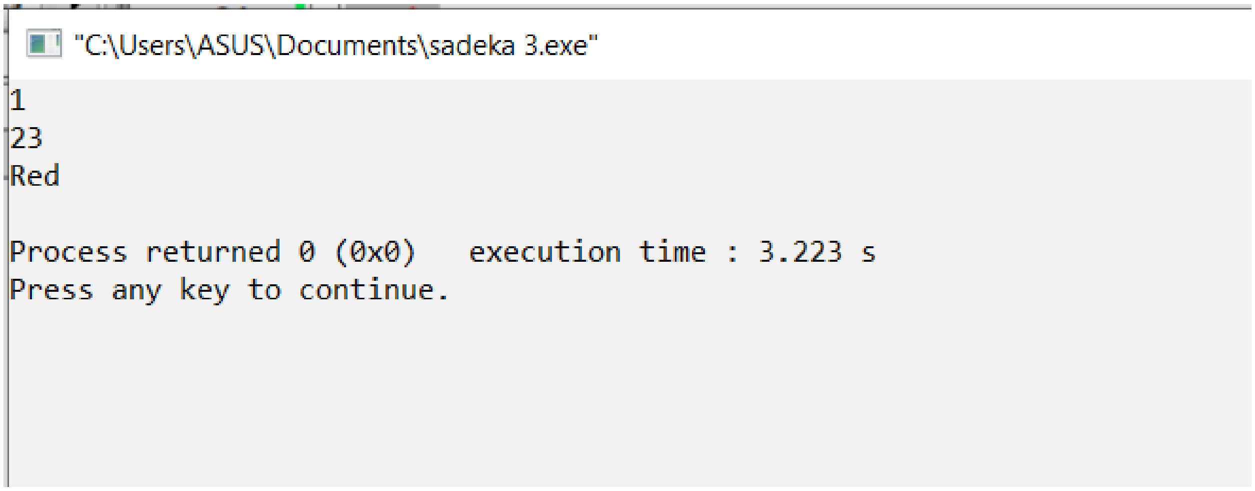
```
Car:
7
2
400
Truck:
4000
32
1200

Process returned 0 (0x0)   execution time : 3.129 s
Press any key to continue.
```

03.

```
#include<iostream>

using namespace std;

class triangle

{ public:

  double side1,side2,side3;

  string color;

  double A,P;

  trianle()

  {

     side1=1.0;

     side2=1.0;

     side3=1.0;

  }
```

```
   triangle(double a, double b,double c)
   {
       side1=a;
       side2=b;
       side3=c;

   }
   accessor(double x, double y)
   { A=x;
   P=y;

   }

   double getArea()
   {
    A =0.5*side1*side2;
       return A;
   }
   double getPerimeter()
   {
    P=side1+side2+side3;
       return P;
   }
void show()
{
    color="Red";
    cout<<color<<endl;
}
```

```
};

int main()
{
 triangle ob(8,6,9);
 double ans=ob.getArea();
 if(ans>0)
    cout<<"1"<<endl;
 else
    cout<<"0"<<endl;
 double ans2=ob.getPerimeter();
 cout<<ans2<<endl;
ob.show();
}
```

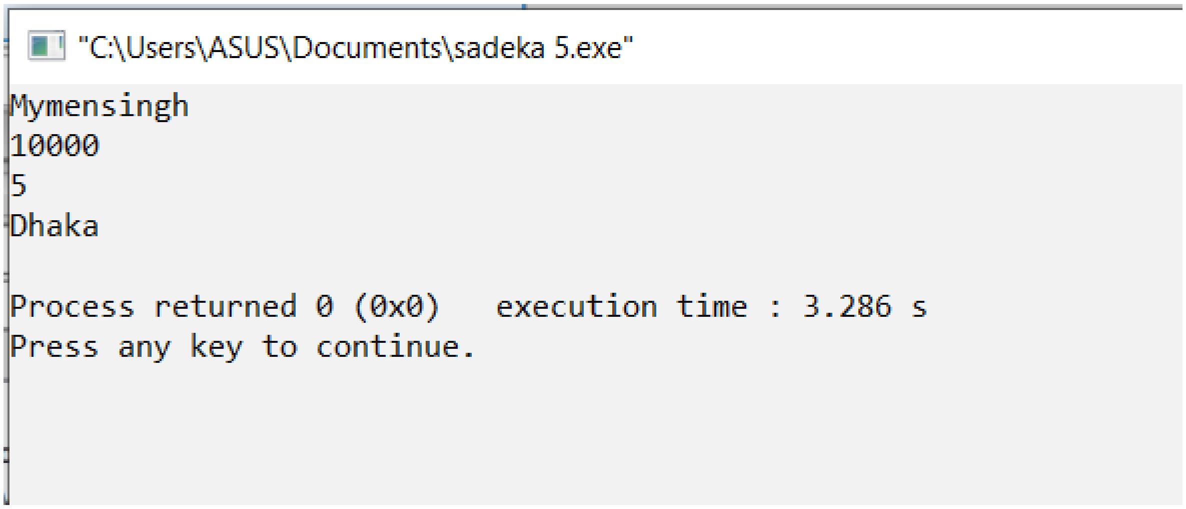


"C:\Users\ASUS\Documents\sadeka 3.exe"

```
1
23
Red

Process returned 0 (0x0)   execution time : 3.223 s
Press any key to continue.
```

04.

```
#include<iostream>
using namespace std;

class division
{
public:
    int num_cities;
    string most_populous_city;
    division(int a, string b)
    {
        num_cities=a;
        most_populous_city=b;
    }
   virtual void show()
    {
        cout<<num_cities<<endl;
        cout<<most_populous_city<<endl;
    }
};
class city:public division
{
public:
    string name;
    int population;
    city(string x, int y,int n,string m):division(n,m)
    {
        name=x;
        population=y;
    }
    void show()
    {
        cout<<name<<endl;
        cout<<population<<endl;
        cout<<num_cities<<endl;
        cout<<most_populous_city<<endl;
    }
};
int main()
{
    city ob("Mymensingh", 10000, 5,"Dhaka");
    ob.show();
}
```



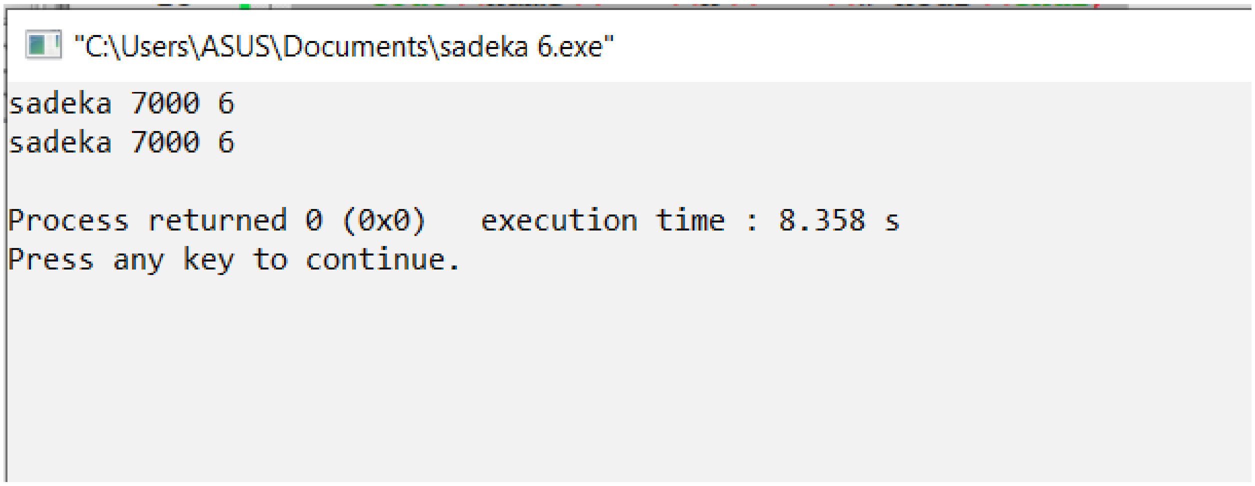
"C:\Users\ASUS\Documents\sadeka 5.exe"

```
Mymensingh
10000
5
Dhaka

Process returned 0 (0x0)   execution time : 3.286 s
Press any key to continue.
```

05.

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
class person
{
public:
    string name;
    long long n,w_hour;
    person()
    {
        cin>>name>>n>>w_hour;
    }
    void display(){
    cout<<name<<" "<<n<<" "<<w_hour<<endl;
    }
};
class academic:virtual public person
{
    public:

};
class non_academic: virtual public person
{
    public:



};
```

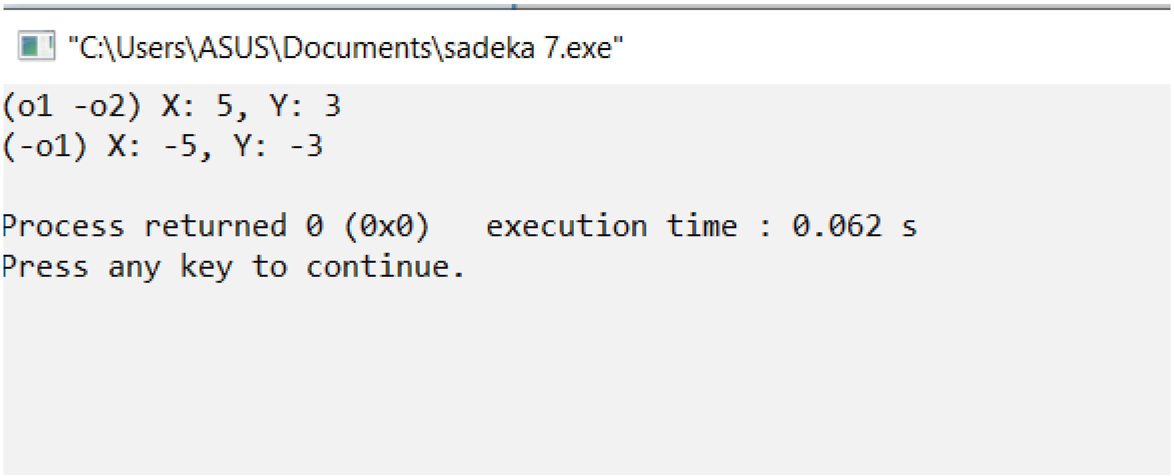
```
class supporting_stuf:public academic,public non_academic
{
   public:

};
int main()
{
   supporting_stuf ob1;
   ob1.display();
}
```



```
sadeka 7000 6
sadeka 7000 6

Process returned 0 (0x0)   execution time : 8.358 s
Press any key to continue.
```

06.

```
# include <iostream >
using namespace std;
class coord
{
int x, y;
public :
coord () { x=0; y=0; };
coord (int i, int j) { x=i; y=j; }
void get_xy (int &i, int &j) { i=x; j=y; }
coord operator -( coord ob2);
coord operator -();
};



coord coord :: operator -( coord ob2)
{
coord temp ;
temp .x = x - ob2 .x;
temp .y = y - ob2 .y;
return temp ;
}

coord coord :: operator -()
{
x = -x;
y = -y;
return * this ;
}

int main ()

{

coord o1 (10 , 10) , o2 (5, 7);

int x, y;

o1 = o1 - o2;

o1. get_xy (x, y);

cout << "(o1 -o2) X: " << x << ", Y: " << y << "\n";

o1 = -o1;

o1. get_xy (x, y);

cout << "(-o1) X: " << x << ", Y: " << y << "\n";

return 0;

}
```

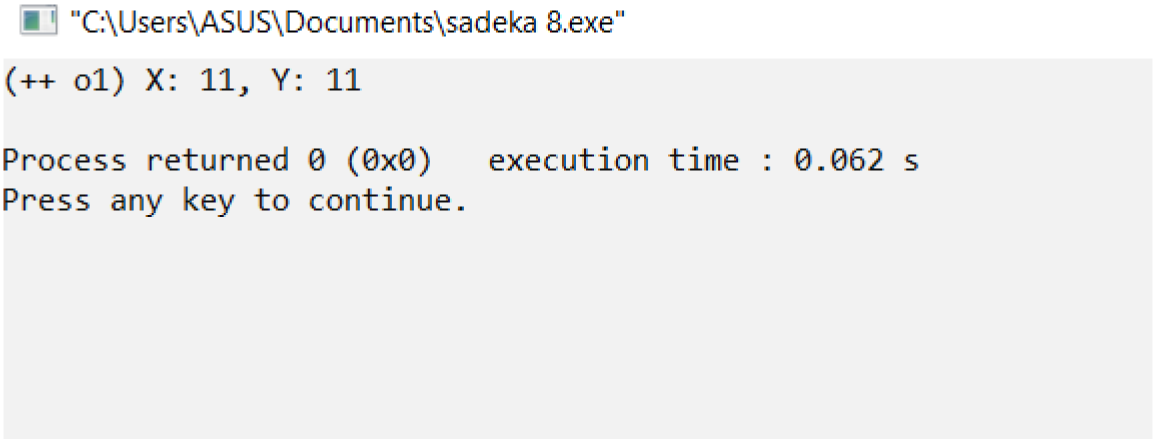


"C:\Users\ASUS\Documents\sadeka 7.exe"

```
(o1 -o2) X: 5, Y: 3
(-o1) X: -5, Y: -3

Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```

07.

```
# include <iostream >
using namespace std;
class coord
{
int x, y;
public :
coord () { x=0; y=0; };
coord (int i, int j) { x=i; y=j; }
void get_xy (int &i, int &j) { i=x; j=y; }
coord operator ++() ;
};
coord coord :: operator ++()
{
x++;
y++;
return * this ;
}
int main ()
{
coord o1 (10 , 10);
int x, y;
++ o1;
o1.get_xy(x,y);
cout << "(++ o1) X: " << x << ", Y: " << y << "\n";
return 0;
}
```



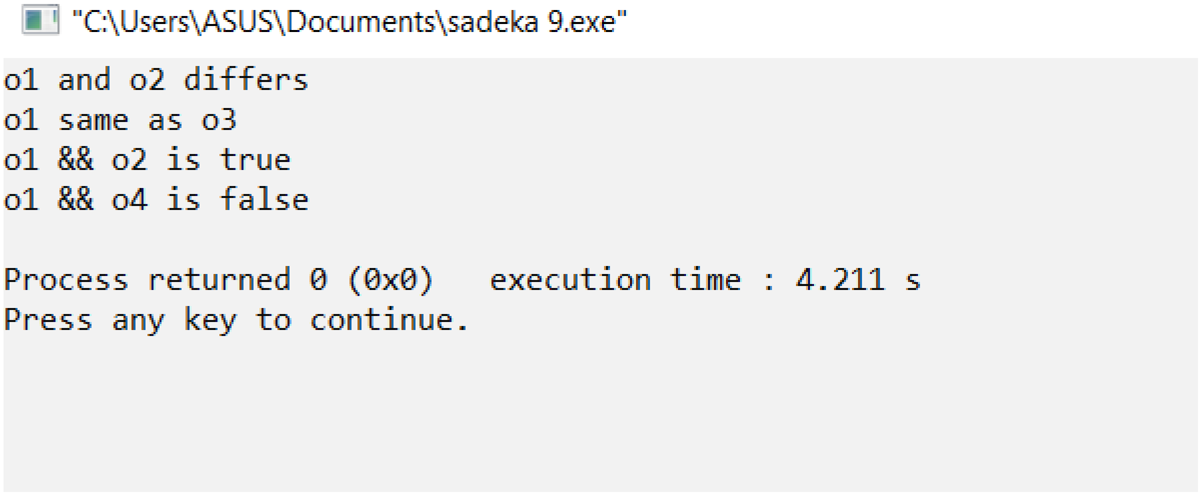
"C:\Users\ASUS\Documents\sadeka 8.exe"

```
(++ o1) X: 11, Y: 11

Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```

08.

```
# include <iostream >

using namespace std;

class coord

{

int x, y;

public :

coord () { x=0; y=0; };

coord (int i, int j) { x=i; y=j; }

void get_xy (int &i, int &j) { i=x; j=y; }

int operator ==( coord ob2);

int operator &&( coord ob2);

};
```

```
int coord :: operator ==( coord ob2)
{
return x== ob2.x && y== ob2.y;
}

int coord :: operator &&( coord ob2)
{
return (x && ob2.x) && (y && ob2.y);
}
int main ()
{
coord o1 (10 , 10) , o2 (5, 3) , o3 (10 , 10) , o4 (0, 0);
if(o1 == o2)
cout << "o1 same as o2\n";
else
cout << "o1 and o2 differs \n";
if(o1 == o3)
cout << "o1 same as o3\n";
else
cout << "o1 and o3 differ \n";
if(o1 && o2)
cout << "o1 && o2 is true \n";
else
cout << "o1 && o2 is false \n";
if(o1 && o4)
cout << "o1 && o4 is true \n";
else
cout << "o1 && o4 is false \n";
return 0;
}
```

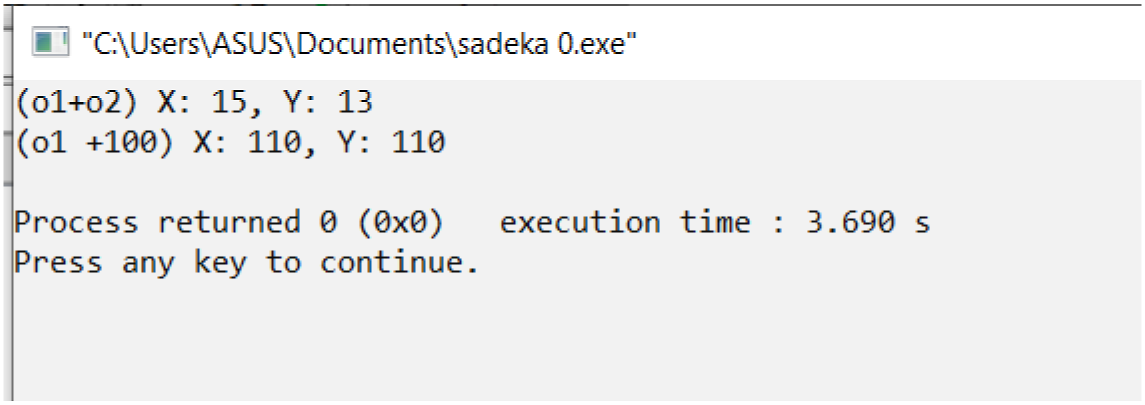


```
o1 and o2 differs
o1 same as o3
o1 && o2 is true
o1 && o4 is false

Process returned 0 (0x0)   execution time : 4.211 s
Press any key to continue.
```

09.

```
# include <iostream >

using namespace std;

class coord

{

int x, y;

public :

coord () { x=0; y=0; };

coord (int i, int j) { x=i; y=j; }

void get_xy (int &i, int &j) { i=x; j=y; }

coord operator +( coord ob2);

coord operator +( int i);

};
```

```
coord coord :: operator +( coord ob2)
{
coord temp ;
temp .x = x + ob2 .x;
temp .y = y + ob2 .y;
return temp ;
}
coord coord :: operator +( int i)
{
coord temp ;
temp .x = x + i;
temp .y = y + i;
return temp ;
}
int main ()
{
coord o1 (10 , 10) , o2 (5, 3) , o3;
int x, y;
o3 = o1 + o2;
o3. get_xy (x, y);
cout << "(o1+o2) X: " << x << ", Y: " << y << "\n";
o3 = o1 + 100;
o3. get_xy (x, y);
cout << "(o1 +100) X: " << x << ", Y: " << y << "\n";
return 0;
}
```



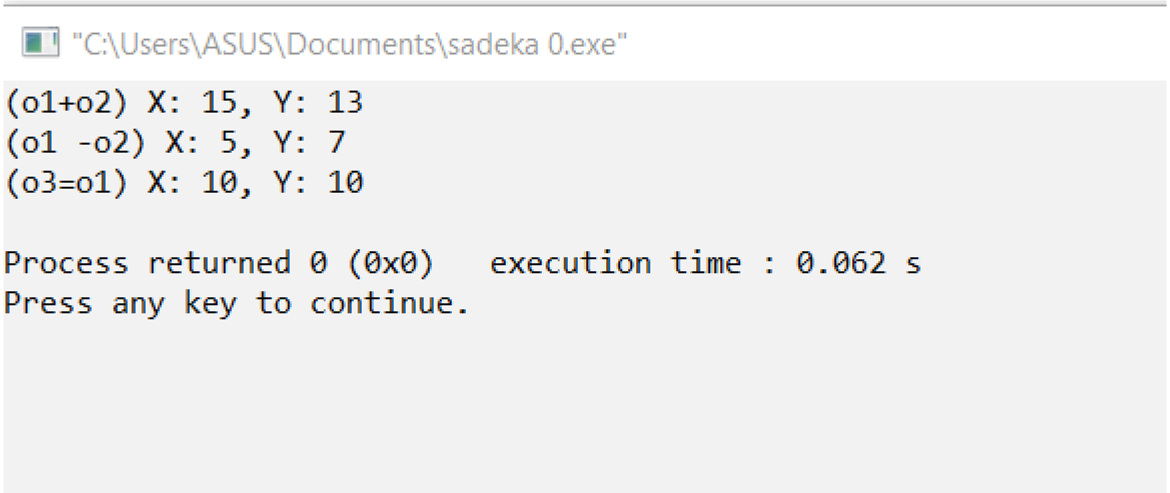
"C:\Users\ASUS\Documents\sadeka 0.exe"

```
(o1+o2) X: 15, Y: 13
(o1 +100) X: 110, Y: 110

Process returned 0 (0x0)   execution time : 3.690 s
Press any key to continue.
```

10.

```
// Overload the +, -, and = relative to coord class .

# include <iostream >

using namespace std;

class coord

{

int x, y; // coordinate values

public :

coord () { x=0; y=0; };

coord (int i, int j) { x=i; y=j; }

void get_xy (int &i, int &j) { i=x; j=y; }

coord operator +( coord ob2);

coord operator -( coord ob2);

coord operator =( coord ob2);

};


// Overload + relative to coord class .

coord coord :: operator +( coord ob2)
```

```
{
coord temp ;
temp .x = x + ob2 .x;
temp .y = y + ob2 .y;
return temp ;
}
// Overload - relative to coord class .
coord coord :: operator -( coord ob2)
{
coord temp ;
temp .x = x - ob2 .x;
temp .y = y - ob2 .y;
return temp ;
}
// Overload = relative to coord class .
coord coord :: operator =( coord ob2)
{
x = ob2.x;
y = ob2.y;
return * this ; // return the object that is assigned
}
int main ()
{
coord o1 (10 , 10) , o2 (5, 3) , o3;
int x, y;
o3 = o1 + o2; // add two objects - this calls operator +
o3. get_xy (x, y);
cout << "(o1+o2) X: " << x << ", Y: " << y << "\n";
o3 = o1 - o2; // subtract two objects
o3. get_xy (x, y);

cout << "(o1 -o2) X: " << x << ", Y: " << y << "\n";

o3 = o1; // assign an object

o3. get_xy (x, y);

cout << "(o3=o1) X: " << x << ", Y: " << y << "\n";

return 0;

}
```



```
(o1+o2) X: 15, Y: 13
(o1 -o2) X: 5, Y: 7
(o3=o1) X: 10, Y: 10

Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```