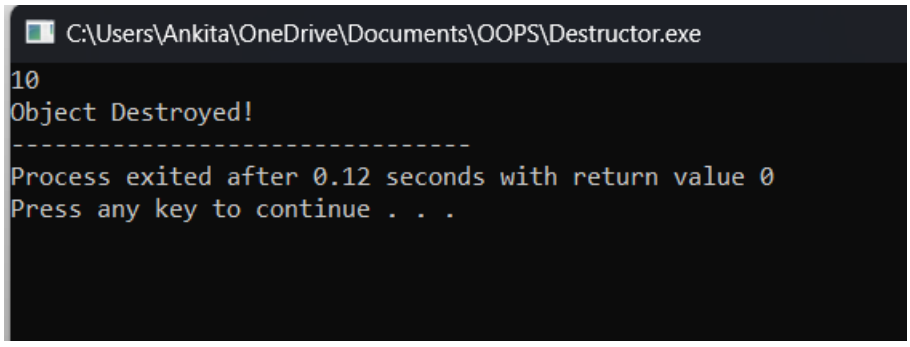
Destructor

Input:

```
#include<iostream>
using namespace std;
class test{
	public:
		test()
		{
			int n=10;
			cout<<n<<endl;
		}
		~test()
		{
			cout<<"Object Destroyed!";
		}
};
int main()
{
	test obj;
	return 0;
}
```

Output:

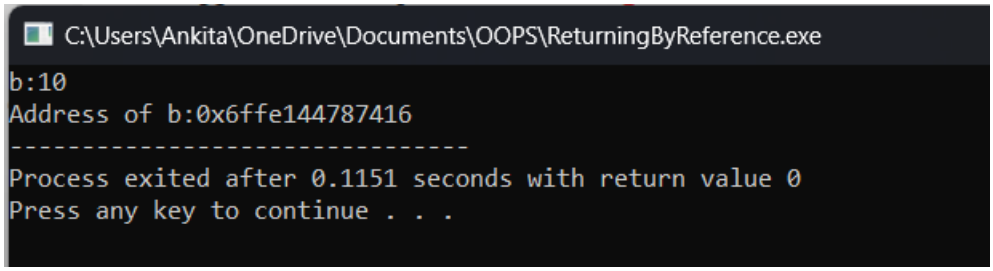


Returning By Reference

Input:

```
#include<iostream>
using namespace std;
class new1{
	public:
		int&fun(int&x)
		{
			cout<<"b:"<<x<<endl<<"Address of b:"<<&x;
		}
};
int main()
{
	int a = 10;
	int &b = a;
	new1 obj;
	cout<<obj.fun(b);
	return 0;
}
```

Output:



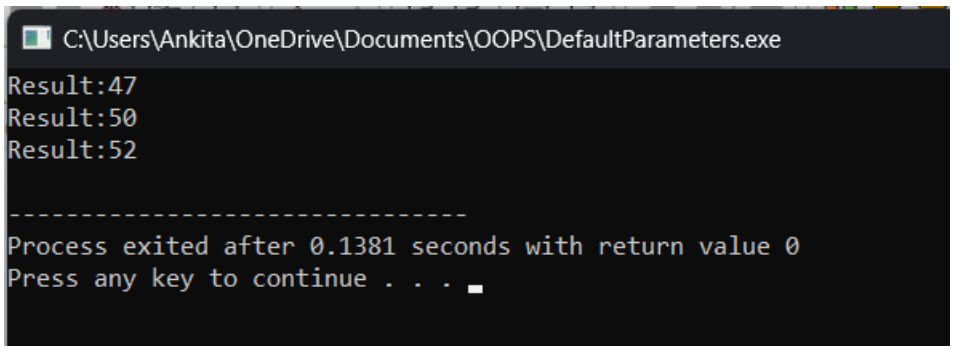
Default Parameter

Input:

```
#include<iostream>
```

```
using namespace std;
class z{
	int res;
	public:
		void add(int a, int b, int c=10){
			res = a + b + c;
		}
		void display(){
			cout<<"Result:"<<res<<endl;
		}
};
int main(){
	z ob1, ob2, ob3;
	ob1.add(17, 20);
	ob2.add(25, 15);
	ob3.add(13, 20, 19);
	ob1.display();
	ob2.display();
	ob3.display();
	return 0;
}
```

Output:



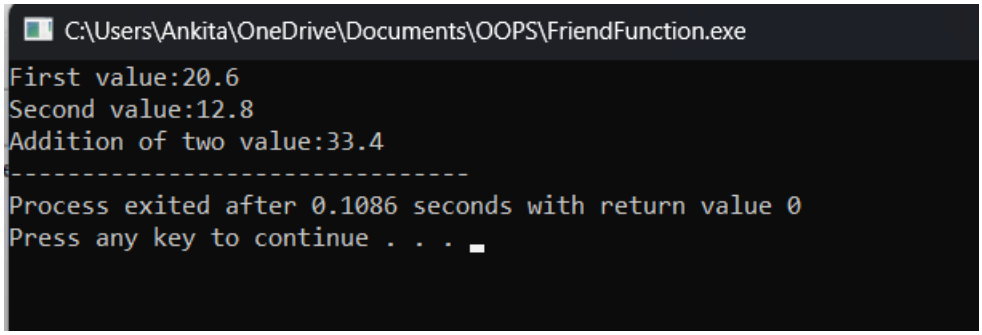
Friend Function

Input:

```
#include<iostream>
using namespace std;
class value{
	float a, b;
	public:
		value1(float x, float y)
		{
			a = x;
			b = y;
		}
		friend value add(value obj);
		void print()
		{
			cout<<"First value:"<<a<<endl;
			cout<<"Second value:"<<b<<endl;
		}
};
value add(value obj)
{
	cout<<obj.a+obj.b;
}
int main()
{
	value v;
	v.value1(20.6, 12.8);
	v.print();
	cout<<"Addition of two value:";
```

```
        add(v);
        return 0;
}
```

Output:



Static Function

Input:

```
#include<iostream>
using namespace std;
class new1{
        float f;
        static int a;
        public:
                void getdata(float x)
                {
                        f = x;
                }
                void showdata()
                {
                        cout<<"f:"<<f<<endl;
                        cout<<"a:"<<a++<<endl;
                }
                static float new2()
```

```
        {
            a = a+1;
            cout<<"a:"<<a<<endl;
        }
};
int new1::a;
int main()
{
    new1 obj;
    obj.getdata(12.6);
    obj.showdata();
    new1::new2();
    obj.showdata();
    return 0;
}
```

Output: