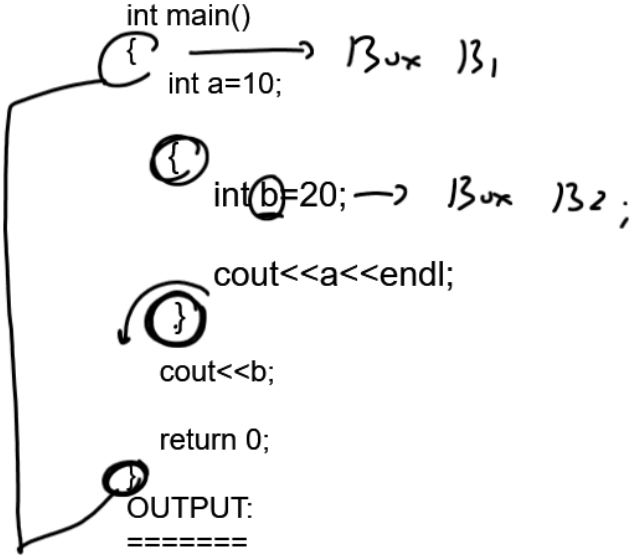
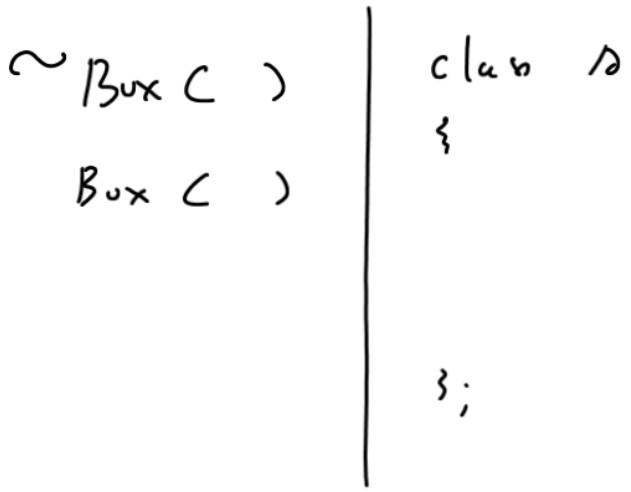
DESTRUCTOR
==========

```
#include <iostream.h>
int main()
{
    int a=10;

    {
        int b=20;

        cout<<b<<endl;
     }
   cout<<a;

   return 0;
}
OUTPUT:
=======
20
10
```

```
#include <iostream.h>
int main()
{
   int a=10;

    {
       int b=20;

       cout<<a<<endl;
    }
   cout<<b;

   return 0;
}
OUTPUT:
=======
20
10
```





Destructor
=========
In C++ a destructor is a special member function of a class which automatically gets call just before the object is to be destroyed. Moreover the name of destructor is also exactly same as the name of the class but it is prefixed with tilde (~). So if the name of the class is **Emp** then its constructor will be **Emp()** and its desctructor will be **~Emp().**

Also we must remember that both constructor and desctructor are automatically called by the C++ compiler. The constructor gets called as soon as the object of a class arrives in memory while the desctructor gets called just before the object is to be removed from memory. And both these calls are automatically given by the C++ compiler.

One more point to remember about desctructor is that if we don't create any desctructor in a class then the C++ compiler itself provides an empty body destructor in the class called as **DEFAULT DESTRUCTOR.**