**Calling Destructors**

**Prerequisite**:

+ Implement default constructor (constructor with no parameter)

+ Implement default copy constructor (copy all memory of an object to other, implemented by system)

+ Implement copy constructor (constructor with one parameter which same type of the object and implemented by user).

+ Applying above operators to declare and assign object.

Ex: 1. objectB b;

b = someFunc();

2. objectB b = someFunc();

In Example 1, default constructor is called, then copy constructor is called to assign object return by someFunc() to b.

In Example 2, only default copy constructor is called.

**Idea:**

For function return ref type: nothing to do to prepare for calling destructors

For function return object type: add address of object to destructor list.

If the object return by the function will be assigned to another object, it is no need to call copy constructor for new object and destructor for returned object.

**Implementation:**

1. Prepare parameter’s space.
2. Add return address to destructor list if the function return an object.
3. If the function is default copy constructor (add address of parameter #2 to destructor optimized map).
4. Rescusive handling destructor calling for each parameter if it is function.
5. Reduce destructor list by checking each address with optimized map. If it is existed in the map, remove from destructor list.