C/C++ 2017/18 programming exercise 4

This exercise is about memory management in C++ in combination with class hierarchies.

The class definitions are here:

`http://www.cs.bham.ac.uk/~hxt/2017/c-plus-plus/destructobj.h`

Your task is to implement appropriate destructors such that deleting the top node of an AST recursively deallocates the whole tree. You may assume that the AST is represented as a tree in memory, so that there is no sharing of nodes. Note that this restriction to trees makes the exercise a great deal easier.

Your code should be in a file `destructobj.cpp`. You should not print anything.

You may use the following includes:

```
#include <string>
using namespace std;
#include "destructobj.h"
```

A minimal main file is here, though you may wish to write your own test cases:

`http://www.cs.bham.ac.uk/~hxt/2017/c-plus-plus/destructobjmain.cpp`

To compile the code, you should use:

```
clang++ -std=c++11 -Wall -Werror -o destructobj destructobj.cpp destructobjmain.cpp
```

Your code must compile on the lab machines as above. If it fails to compile or produces any errors, it will be given 0 marks.

Marking scheme

This exercise counts for 5% of the module mark.

Valgrind should not report anything when used like this:

```
valgrind -q --leak-check=full ./destructobj
```