EE3093 Tutorial: C++ week 3

Instructions to users

This tutorial can be completed up to 2 degrees of complexity (one for each section); make sure you have understood and completed a section (i.e. code implemented and tested successfully) before moving on to the next one.

Part 1: Basic

Follow the instructions given in the second Tutorial and create (in a new directory) a new project (e.g. TUT3).

Copy to the TUT3 folder any header/cpp file (supplied or created by you for/during any past tutorial or lab) that you find required/helpful for this tutorial; then "Add each item to the current Project" (as per instructions in TUT2).

Modify class "rectangle" **AS SHOWN IN LECTURES THIS WEEK**, so that it supports three **static member variables** (integers) used as counters for all rectangle instances:

rectangle_instances_created_count;

rectangle_instances_alive_count;

initialized_rectangle_instances_count;

Also, each rectangle instance should contain the (protected integer) member variable **rectangle_ID** that corresponds to the value of "rectangle_instances_created_count" at the moment of instantiation of the object. The content (value) of this variable can be obtained via the following (public) rectangle member function:

int getRectangleID();

Finally, the class should support a static (void) function **printRectangleCount()** that reports (on screen) the following:
(**1**) the number of Rectangles that have been instantiated up to that point;
(**2**) the number of Rectangles that are currently "active" (i.e. have not been out of scope);
(**3**) the number of Rectangles that are (currently active and) initialized correctly (with positive side values).

Test your implementation **with the following routine**:

void **test_static_counters**()

```
{
	rectangle::printRectangleCount();
	rectangle testobj1;
	rectangle::printRectangleCount();
	cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
	testobj1.inputRandomSides();
	rectangle::printRectangleCount();
	{
		cout << endl << "Entering a local scope:"<< endl;
		const int localaraysize=5;
		rectangle testobjarray[localaraysize];
		rectangle::printRectangleCount();
		for(int i=0; i<localaraysize; i++)
		{
			cout << "testobjarray[" << i << "] ID: " << testobjarray[i].getRectangleID() << endl;
			testobjarray[i].inputRandomSides();
			rectangle::printRectangleCount();
		}
		cout << "Exiting local scope;"<< endl << endl;
	}
	rectangle::printRectangleCount();
	testobj1.reset();
	rectangle::printRectangleCount();
}
```

Part 2: Intermediate

Apply a similar modification to class **square** (implemented in TUT2). This class should include static member variables that implement counters analogous to the ones introduced before for class rectangle (also a **square_ID**, uniquely assigned to each instance of the class square when the object is instantiated, analogously to the **rectangle_ID**); likewise, it includes a static member function **printSquareCount()** similar to the .

To test your implementation, write & run a routine analogous to the one reported in the previous section.

If your implementation of “square” incorporates a “rectangle”, every instance of “square” should increment the count of instantiated “rectangles” (this is acceptable: a square is a particular rectangle). Test this by reporting the value of the rectangle (static) counters immediately after you report those relative to class square.