Writing tests for functions:

Here's a table that tells you what you need to do to get points for testing different types of functions:

	function that does not return	**function that returns values**
0 parameters	call this function 1+ times in your main function	call this function 1+ times in your main function and cout the return value.
1+ parameters	call this function 2+ times in your main function with different parameters	call this function 2+ times in your main function with different values and cout the return values

On the following page there are example functions with test cases.

If you have any questions, please ask on Piazza!

	function that does not return	function that returns values

0 parameters	#include <iostream>
	using namespace std;
	
	void sayHello()
	{
	    cout << "Hello world" << endl;
	}
	
	int main()
	{
	    // test 1
	    // expected output
	    // Hello world
	    sayHello();
	}
		#include <iostream>
		using namespace std;
		
		int getSecondsInDay()
		{
		    return 86400;
		}
		
		int main()
		{
		    // test 1
		    // expected output
		    // 86400
		    int seconds = getSecondsInDay();
		    cout << seconds << endl;
		}

1+ parameters	void sayHello(string name)
	{
	    cout << "Hello " << name << endl;
	}
	
	int main()
	{
	    // test 1
	    // expected output
	    // Hello Bob
	    sayHello("Bob");
	
	    // test 2
	    // expected output
	    // Hello Mary
		#include <iostream>
		using namespace std;
		
		float calcPay(float hours, float pay_rate)
		{
		    float total_pay = hours * pay_rate;
		    return total_pay;
		}
		
		int main()
		{
		    // test 1
		    // arguments: hours = 14, pay_rate = 12
		    // expected output: pay1=480
		    // explanation: 14 * 12 = 480

	sayHello("Mary");
	}
		float pay1 = calcPay(40, 12);
		    cout << "pay1=" << pay1 << endl;
		
		    // test 2
		    // arguments: hours = 30, pay_rate = 20
		    // expected output: pay2=600
		    // explanation: 30 * 20 = 600
		    float pay2 = calcPay(30, 20);
		    cout << "pay2=" << pay2 << endl;
		}