Lab Document and Questions
Name: _____

Lab #12, 11/21/2013
WFU Username: _____

CSC 111E: Lab #12 – Classes and Objects

Lab Date: Thursday, 11/21/2013 Due Date: Friday, 11/22/2013 @ 5:00pm

Purpose: The purpose of this lab is to have you gain experience in using classes and objects in Visual Basic.

Program 1: Designing a class to be used
Design a class that represents a StockSale (the sale of a stock). Your class should include the following variables and methods:

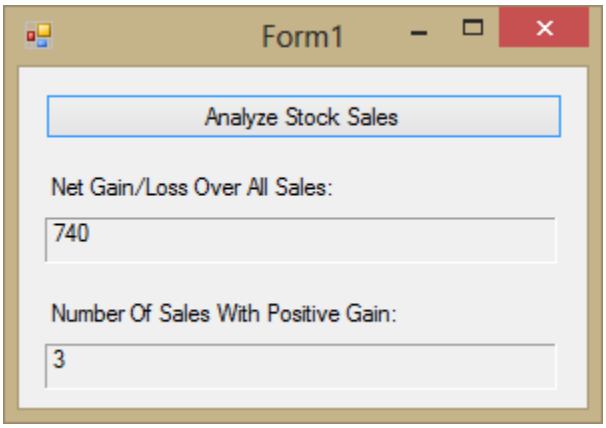
Variables:

- A variable representing the stock name (a String)
- A variable representing the gain for the sale (a double) [Sometimes you lose money on selling a stock, a a negative gain will represent a loss]

Methods:

- A constructor (the "New" method) which sets the stock name to "" (the empty string) and gain to 0.0
- A getStockName function that returns the name of the stock
- A setStockName procedure that allows the stock name to be changed
- A getGain function that returns the gain amount
- A setGain procedure that allows the gain amount to be changed
- A positiveGain function that returns true if the gain is > 0 and false otherwise.

I have written a Form1.vb which will make use of your class and report various information about stocks. This form and a data file are available to you as part of *Lab12Program1.zip* in Sakai. If your code implementing the class is done correctly, you should see outputs shown at right when you run the program and press the Analyze button.



Program 2: Using classes
In several examples and practice problems we have looked at this semester, the notion of "random numbers" has been useful. In particular, we explored this in a number of scenarios where we needed to simulate a system in action (customers entering a store and being handled, birth rates, …). This problem will have you make use of a Die class and a Simulation class. The Die class is supposed to represent a die, like you would roll in Monopoly. The Simulation class is going to help us keep track of how often values occur from rolls of two dice (so it will help us keep track of how often, when rolling two dice that we roll a sum of 2, a sum of 3, …, up to a sum of 12). Here are the public methods you can make use of in the Die and Simulation classes. These are already written for you – you will just make use of them.

Die:
New() – This method (a procedure) is the constructor for the Die class
Roll() – This method (a function) returns an integer between 1 and 6 (as you would get from rolling a die).
Simulation:
New() – This method (a procedure) is the constructor for the Simulator class and sets all counts for each possible sum to 0
GetCountFor(int value) – This method (a function) returns how many times we have seen a particular sum
IncrementCountFor(int value) – This method (a procedure) adds 1 to the number of times we have seen a particular sum

Download *Lab12Program2.zip* from Sakai to get the implementations of the two classes above and the GUI you will work with in this program. The GUI, written for you and shown below, has a button that says "Run Simulation". When that button is pressed, you should have code that does the following:

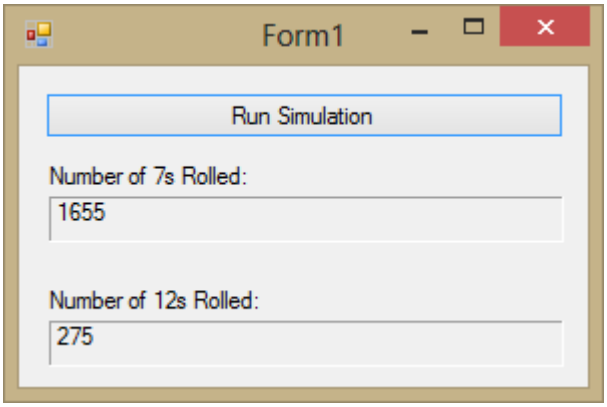
Create variables representing an instance of a Die (a Die object) and an instance of a Simulator (a Simulator object)

Run 10000 trials where you:

- Roll the die twice
- Sum the results of the two rolls
- Increment the count for that sum in your Simulator object

After the 10000 trials end, report into the appropriate output labels the number of rolls of the dice where you saw a sum of 7 and the number of rolls of the dice where you saw a sum of 12. You should get these counts from the Simulator

The expectation is that about 16.7% (1/6th) of your two-die sums should be a 7, and about 2.7% (1/36th) of your two-die sums should be a 12. The numbers won't exactly match what you see below, but should be close.



Submission

To submit this lab for grading, do the following by Friday, 11/22, at 5:00pm – zip your projects and upload both files into Saki (under Assignments, Lab 12). See next page for grading information.

Your grade will be based on the following rubric:

Objective	Points Available	Earned
Problem 1: StockSale class supports both requested variables	8	
Problem 1: StockSale class supports all six requested methods	24	
Problem 1: Public/private used correctly for StockSale class	8	
Problem 1: StockSale class correctly implements all six requested methods	24	
Problem 2: Program employs Die and Simulator classes to come up with answer	18	
Problem 2: Program comes up with correct answers	18	