Lab 9 Instructions: Methods (Group B)

CS133, Beginning Programming: C#

Introduction

The objective of this lab is to give you practice defining and calling methods.

Assignment

Part 1: Chapter 6, Modularizing Your Code with Methods – Tutorials

Complete tutorials 6-3 through 6-5, at the end of chapter 6 in the textbook. (The starting files for chapter 6 are in the source code provided with the textbook.)

Part 2: Chapter 6 Programming Problems:

2. Falling Distance
When an object is falling because of gravity, the following formula can be used to determine the distance the object falls in a specific time period:

$$d = 1/2\ g\ t^2$$

The variables in the formula are as follows: d is the distance in meters, g is 9.8, and t is the amount of time in seconds that the object has been falling. Create an application that allows the user to enter the amount of time that an object has fallen and then displays the distance that the object fell. The application should have a function named FallingDistance. The FallingDistance function should accept an object's falling time (in seconds) as an argument. The function should return the distance in meters that the object has fallen during that time interval. (Gaddis 384)

4. Calories from Fat and Carbohydrates
A nutritionist who works for a fitness club helps members by evaluating their diets. As part of her evaluation, she asks members for the number of fat grams and carbohydrate grams that they consume in a day. Then, she calculates the number of calories that result from the fat using the following formula:

Calories from fat = Fat grams × 9

Next, she calculates the number of calories that result from the carbohydrates using the following formula:

Calories from carbs = Carbs grams × 4

Create an application that will make these calculations. In the application, you should have the following methods:

- FatCalories–This method should accept a number of fat grams as an argument and return the number of calories from that amount of fat.
- CarbCalories–This method should accept a number of carbohydrate grams as an argument and return the number of calories from that amount of carbohydrates.

Lab 9 Instructions: Methods (Group B)

CS133, Beginning Programming: C#

7a. Present Value

Suppose you want to deposit a certain amount of money into a savings account and then leave it alone to draw interest for the next 10 years. At the end of 10 years you would like to have $10,000 in the account. How much do you need to deposit today to make that happen? You can use the following formula, which is known as the present-value formula, to find out:

$$P = F / (1 + r)^n$$

The terms in the formula are as follows:

- P is the present value, or the amount that you need to deposit today.
- F is the future value that you want in the account. (In this case, F is $10,000.)
- r is the annual interest rate.
- n is the number of years that you plan to let the money sit in the account.

Write a method named PresentValue that performs this calculation. The method should accept the future value, annual interest rate, and number of years as arguments. It should return the present value, which is the amount that you need to deposit today. Demonstrate the method in an application that lets the user experiment with different values for the formula’s terms. (Gaddis 386 - 387)

7b. PresentValue with *out* parameters

Modify the Present Value problem so that the PresentValue method checks the values passed in to be sure they aren’t negative- those would be invalid inputs. Add two *out* parameters to the method. The first *out* parameter will be a *bool* that is set to true if the input is invalid and the second *out* parameter will be a *string* that holds an error message. The code that calls the PresentValue method will display an error message when the value passed to the PresentValue method is invalid.

Gaddis, Tony. Starting out with Visual C# 2012, 3rd Edition. Pearson, 20130528. VitalBook file.

Files to Submit to Moodle

- A document containing screenshots of the applications you wrote for the Programming Problems running (label each screen-shot).
- Zipped VS solution folders for each problem.
- The completed code review form for your lab work.
- A copy of the code review you gave your code review partner.