CISS 160 Homework – Methods A

The grade you earn will be based on a number of factors, including overall quality/correctness/validity of your application, following the file naming guidelines, following the compressed folder guidelines (see below and Syllabus), and following the other guidelines such as comments for your name, programming comments, test cases etc.

General guidelines for each program:

> Include three comment lines within each program's Form1.cs file with your name, student id number, date, and goal/purpose of the program:

//Author: Your NAME

//ID: Your Student ID Number (NOT YOUR SOCIAL SECURITY NUMBER)

//Date:

//Goal-Purpose of the Program: (…your description…)

> within any program file where you write source code, include comments throughout your code describing in your own words, what the various sections of your code are doing; single line comments can be preceded by two forward slashes //COMMENT…

> for each program create a Text file in the folder where your project files are located and call it TestPlan.txt – edit that file for each program below to include a description of the different 'Tests' that you performed to give yourself assurance that your program is valid/is working correctly; this can include validating data to be numeric, validating that required data was entered, exception handling, other tests to verify specific calculations, functionality, verifying User Interface events etc.

Write the C# programs described below:

> the requirements for each program will be described as a paragraph below OR the requirements will be represented as a compiled .exe of a program (note: this is not source code) that I include in the assignment folder so you can see a running version of how your program should work

> we will cover general aspects of various programming elements during lectures, but it is up to you to combine the lecture learning components along with the textbook reading to create a specific solution

> any images/graphics that are needed will be included in the assignment folder or I will provide in Angel

(see next page)

CISS 160 Homework – Methods A

1. (**75** Points) Create your own Method in a Program in C#. Create a C# windows form that demonstrates how to modularize code into a Method. Be sure to refer to the general programming guidelines listed above. Name your project:

BYOMethodWithReturn - Your Name

THINK: Building Blocks… get incremental parts of the logic working and then keep adding incremental functionality until you have it all done… break it down! AVOID the 'Big Bang' - trying to code it all-at-once approach!

Create a Method that will accept multiple arguments/parameters, and will return a decimal value

A) Start with your previously created Car Cost Comparison Program where you compared various costs of two cars

- NOTE: if you had not completed the original Car Cost Compare Homework assignment then you will need to create it for this assignment
- NOTE: if you completed the original Car Cost Homework, and if I wrote comments about the code you turned in, then you will need to fix your code related to those comments as part of this assignment

B) Other than correcting issues mentioned above, leave your Car Cost Comparison code the same, EXCEPT:

- if it is not already present in your code, add in validation and error messaging for any Text Boxes on the Form that accept numeric data, since at the time the earlier Homework was due, it may not have required validation
- for the calculation labeled 'Calculated 1 Year Total Fuel Cost', replace the calculation code in your current _Click event with a call to a NEW Method that you will create that will do the same
- call your new method YourNameMETHODCalc1YearFuelCost
- your new method's job is to: Calculate the cost of 1 Year of Fuel based on overall Miles Driven Per Year assumption, Fuel Cost Per Gallon assumption and the MPG value (MPG is for one car at a time)
- your new method should be setup as follows:

…

private decimal YourNameMETHODCalc1YearFuelCost (decimal yourMileDrivenPerYearDec variable, decimal yourFuelCostPerGallonDec variable, decimal yourMPGforTheCar variable)

{

// **... for each variable in the method header above, inside the parameter parentheses, use your own variable names and make them be slightly different than the names of the corresponding variables already used in your _Click event code**

// **... for example if your _Click event code currently uses a variable named myMilesDrivenDec, then make the parameter names in the method header be passIN_MilesDrivenDec or local_MilesDrivenDec**

// **… add in your code to declare a local decimal variable within this Method to hold the 1 Year Fuel Cost calculation, call it local_1YearFuelCostDec …**

// **... add in your code to calculate the 1 Year Fuel cost using the variables passed in to the Method...**

// **... add in your code to 'return' the 1 Year fuel cost to the 'caller'**

// **… via the C# return command combined with the 1 Year Fuel Cost variable from this Method...**

}

(see next page)

C) Then, back in your existing _Click event code, change the code that currently calculates the 1 Year Fuel Cost in your _Click event to now 'call' the new method to obtain the 1 Year Fuel Cost FOR EACH OF THE TWO CARs

for example (you will need to fill in the ?? marks with your variable arguments/parameters)

//**.... in your _Click event code call the new method a) for Car1 then b) for Car2 …**

my1YearFuelCostCar1 = YourNameMETHODCalc1YearFuelCost (??, ??, ??);

my1YearFuelCostCar2 = YourNameMETHODCalc1YearFuelCost (??, ??, ??);

//**....**

The above is just an example, your existing code may look different than the above.

All other code should remain the same.

2. (**50 OPTIONAL EXTRA CREDIT POINTS**) Create your own Method in a Program in C#. Create a C# windows form that demonstrates how to modularize code into a Method. Be sure to refer to the general programming guidelines listed above. Make a copy of the above project and Name your project:

BYOMethodWithParmOut - Your Name

THINK: Building Blocks… get incremental parts of the logic working and then keep adding incremental functionality until you have it all done… break it down! AVOID the 'Big Bang' - trying to code it all-at-once approach!

Create a Method that will accept multiple arguments/parameters, and will utilize an 'out' parameter to return a value. An out parameter is considered passing by 'reference' in that the variable being passed in will actually be updated by the method (versus passing by 'value' which only passes a value to the method and the method utilizes the value but does not change it).

After making a copy of the above Part 1 project and naming it as requested…

A) leave your Car Cost Comparison code the same, EXCEPT:

- for the calculation labeled 'Calculated 1 Year Total Fuel Cost', replace the calculation code in your current _Click event with a call to a NEW Method that you will create that will do the same
- comment out the method from the assignment above so that it does not get confused with the below
- call your new method YourNameMETHODCalc1YearFuelCost**Part2**
- your new method's job is to: Calculate the cost of 1 Year of Fuel based on overall Miles Driven Per Year assumption, Fuel Cost Per Gallon assumption and the MPG value (MPG is for one car at a time)

- your new method should be setup as follows:

…

private void YourNameMETHODCalc1YearFuelCostPart2 (decimal yourMileDrivenPerYearDec variable, decimal yourFuelCostPerGallonDec variable, decimal yourMPGforTheCar variable, *out decimal your1YearFuelCost variable*)

{

// ... for each variable in the method header above, inside the parameter parentheses, use your own variable names and make them be slightly different than the names of the corresponding variables already used in your _Click event code

// ... for example if your _Click event code currently uses a variable named myMilesDrivenDec, then make the parameter names in the method header be passIN_MilesDrivenDec or local_MilesDrivenDec

// … for the 'out' variable in the method header prefix it with something like 'out_1YearFuelCost'

// ... add in your code to calculate the 1 Year Fuel cost using the variables passed in to the Method as input and storing the calculation in the 'out' variable named in the header...

// .. no return statement is needed since we have an 'out' variable instead

}

B) Then, back in your existing _Click event code, change the code that currently calculates the 1 Year Fuel Cost in your _Click event to now 'call' the new method to obtain the 1 Year Fuel Cost FOR EACH OF THE TWO CARs

for example (you will need to fill in the ?? marks with your variable arguments/parameters)

//.... in your _Click event code call the new method a) for Car1 then b) for Car2 …

YourNameMETHODCalc1YearFuelCostPart2 (??, ??, ??, *out* ??);

YourNameMETHODCalc1YearFuelCostPart2 (??, ??, ??, *out* ??);

//....

The above is just an example, your existing code may look different than the above.

All other code should remain the same.

Take all of your completed project folders above and copy them into a folder named:

MethodsA - Your Name

Zip up this folder and submit the zip file in this dropbox by the due date and time.