ASSIGNMENT 5

INTRODUCTION

This will be your second non-Bowers assignment. Before you begin this assignment, make sure your **BowersHeading** file has been renamed to **LastNameHeading** (where *LastName* is your own last name).

ASSIGNMENT

Part 1

1. Create a new project named **LastName_5** (where *LastName* is your own last name).
2. Add your **LastNameHeading** file to the project.

Part 2

1. Create a new class named *Person*. In the class:
    a. Create a setter-getter property for WeightInPounds (double)
    b. Create a setter-getter property for HeightInInches (int)
    c. Create a getter for BMIValue.
        i. When called, the BMIValue property will return BMI as a double using the formula for calculating BMI shown in Exercise 3.31 on page 104 of your textbook.
    d. Create a getter for BMIName.
        i. When called, the BMIValue property as a string that is one of the following four values, “Underweight”, “Normal”, “Overweight”, or “Obese.” Use the information shown on page 105 of your text to determine which BMI numeric value corresponds to a BMI text value.
    e. The *Person* class does not require a constructor, however, you may write one if you wish.

Continued on the next page.

Part 3

1. In your main program, you will call the *getHeading* method of your LastNameHeading, passing *Assignment 5* to the method. Then, display a menu a shown in Figure 1 (below).

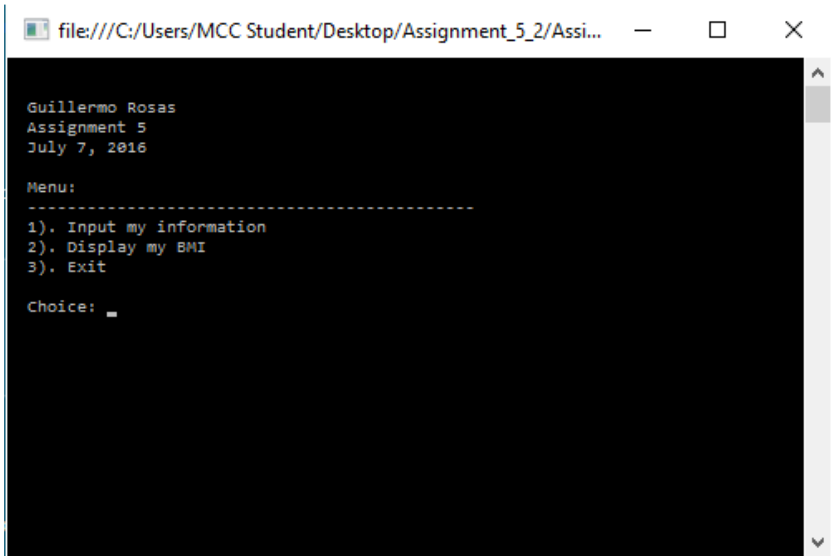

Figure 1

2. The user must enter only the values 1, 2, or 3:

   a. The user must input their information by pressing 1 **before** they can display their BMI.

   b. If the user presses 2 to display BMI *before* they have input their information, you should generate and display an error. A sample error is shown in Figure 2 (below).

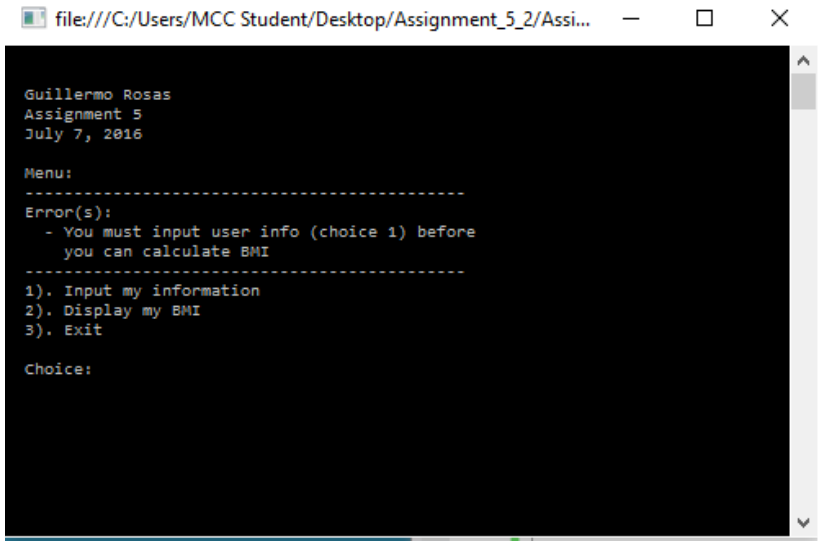

Figure 2

Continued on the next page.

3. When the user presses 1:
    a. The application should prompt the user to input their height in inches and their weight in pounds.
        i. The input must be a positive number greater than zero:
            1. If the user inputs text (e.g., 'cat'), display an error and prompt the user to input a numeric value.
            2. If the user inputs a negative value or inputs zero, display and error and prompt the user to input a numeric value greater than zero.
        ii. Please refer to the sample solution in Blackboard for a sample of these error message. You must display and error and prevent the application from continuing until valid input has been entered; however, you may choose how you display the errors and the text of the errors.
        iii. After the user has input valid height and weight information, return the user to the main menu.
4. When the user presses 2:
    a. Create an instance of the *Person* class, passing the height and weight information to the appropriate properties (depending on how you solve this exercise, you may have accomplished this before the user presses 2, that's OK – this is just where I'm putting the instruction).
    b. Display the BMI Values table shown on page 105 of your textbook, then, display the individual's BMI value from the instance of the *Person* class (round the result to two decimal places) then display the text value of their BMI (BMIName). Finally, call LastNameHeading.getClosing(). Figure 3 (below) shows an example of how this output might appear.
    c. When the user presses **ENTER**, they should return to the menu.

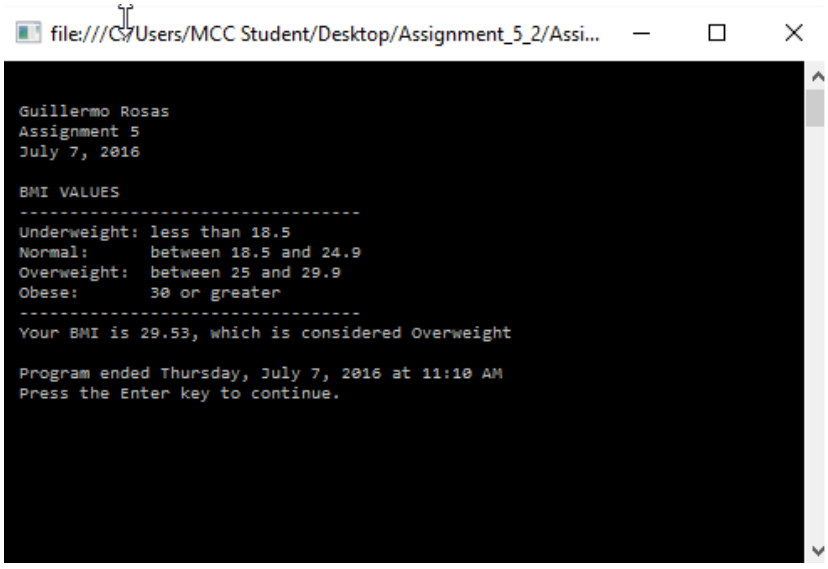

Figure 3

Continued on the next page.

5. When the user presses 3:

    a. Display a nice message and your personal heading's closing.

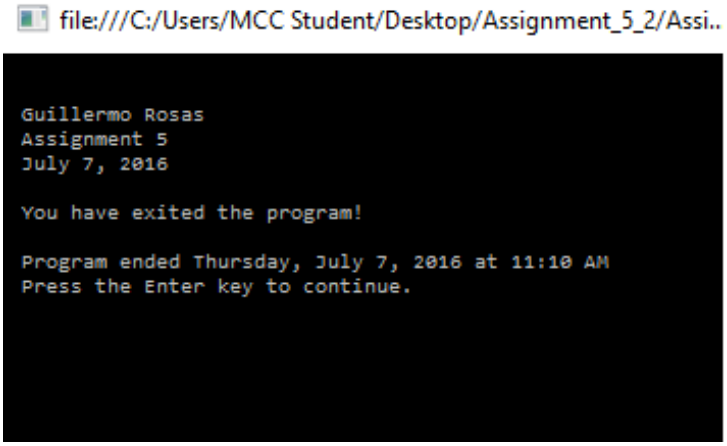

Figure 4

OTHER NOTES

- Each screen should display your personal heading (LastNameHeading.getHeading("Assignment 5") – the included sample may be missing the heading on some screens; however, make sure it's there on yours.
- This assignment is a bit free form; however, when determining the BMIName, make sure to use *Logical Operators* and either ifs, or if-else as you see if.
- Formatting is largely up to you; however, all output must be two spaces from the left, to align with the header.
- You only need to call LastNameHeading.getClosing() when you display the BMI output or when the user has opted to exit by pressing "3."
- This is a bullet just to make David Ream feel comfortable with the assignment.
- To determine if input is a number: you will have to do some research, but take a look at the TryParse() method in your number classes (Double, Int32, etc.) and do some sleuthing in the MSDN documentation.
- How can you track errors? Good question. Research some ways.
- The menu must display using a loop structure. The only time your application will exit is if the user enters "3" at the menu.

Continued on the next page.

- Avoid code repetition. If you find you're repeating code, write functions. Keep in mind, if you're writing functions in your main program class, because your main function is a static function, all functions within that class will also have to be static (this will not affect the Person class). Here are some examples of how you can write a static function that returns no value, and a static function that returns a string:

```
static void MyFunction() {
      //Body here
}

static string MyOtherFunction() {
      //Body here
}
```

COMPLETION

- When complete, please ZIP the entire solution into a single file named **LastName_5.zip** (where *LastName* is your own last name).
- Upload and submit the ZIP file to the Assignment 5 drop box in Blackboard.

Fin.