ASSIGNMENT 6

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).

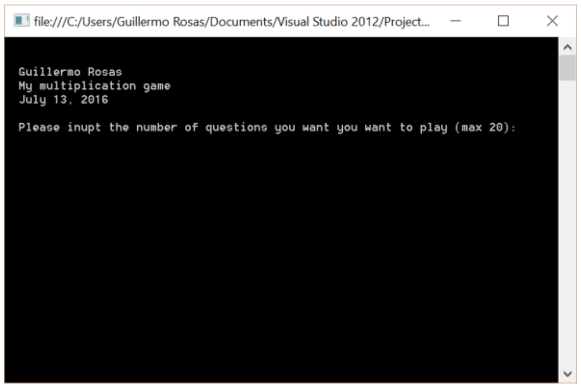
ASSIGNMENT

Part 1

1. Create a new project named **LastName_6** (where *LastName* is your own last name).
2. Add your **LastNameHeading** file to the project.

Part 2

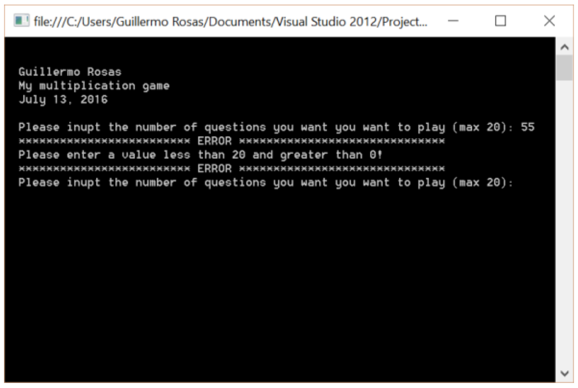
1. Complete exercises 7.39 on page 284 of your textbook. Make the following modifications to your program:
    a. Your program should display a "start" screen.
        i. The top of the start screen should display your personal heading (LastNameHeading.getHeading()). In this exercise, you'll use the .getHeading() method to display the name of your "game." If you're not creative, name your game "Assignment 6" – if you are creative, give it a nice name.
        ii. Prompt the user to select the number of questions they wish to play (max 20):
            1. Verify the input is a number, if the input is not a number, display an error and prompt the user to input the number of questions they want to play.
            2. Verify the input is no less than zero and no greater than 20. If the user inputs a value that is less than zero or greater than 20, display an error and prompt the user to input the number of questions they want to play. Figure 1 (below) shows an example of what this should look like.



```
file:///C:/Users/Guillermo Rosas/Documents/Visual Studio 2012/Project...

Guillermo Rosas
My multiplication game
July 13, 2016

Please inupt the number of questions you want you want to play (max 20):
```



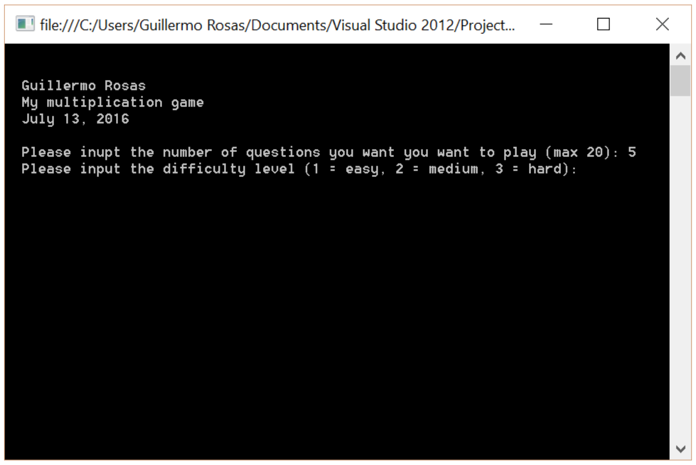
```
file:///C:/Users/Guillermo Rosas/Documents/Visual Studio 2012/Project...

Guillermo Rosas
My multiplication game
July 13, 2016

Please inupt the number of questions you want you want to play (max 20): 55
************************* ERROR ******************************
Please enter a value less than 20 and greater than 0!
************************* ERROR ******************************
Please inupt the number of questions you want you want to play (max 20):
```

Figure 1

iii. If the user input is valid, prompt the user to select a difficulty level (1 = easy, 2 = medium, 3 = hard). Verify the user input is a numeric value, and verify the user input is 1, 2, or 3 only. If the user inputs erroneous data, display an error and prompt the user to input a difficulty level.

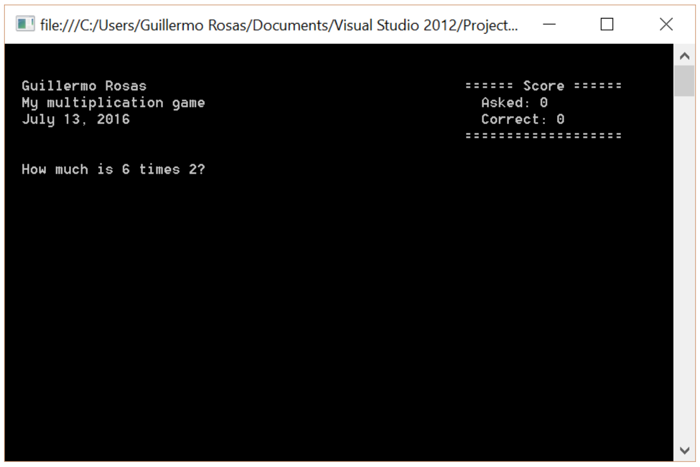

file:///C:/Users/Guillermo Rosas/Documents/Visual Studio 2012/Project...

```
Guillermo Rosas
My multiplication game
July 13, 2016

Please inupt the number of questions you want you want to play (max 20): 5
Please input the difficulty level (1 = easy, 2 = medium, 3 = hard):
```

Figure 2

iv. Clear the screen and display your heading again, with your game title. Also, display a "scoreboard" in the upper right hand corner of the console window. This may take a little research to figure out how to do this.


file:///C:/Users/Guillermo Rosas/Documents/Visual Studio 2012/Project...

```
Guillermo Rosas                                            ====== Score ======
My multiplication game                                       Asked: 0
July 13, 2016                                                Correct: 0
                                                           ==================

How much is 6 times 2?
```

Figure 3

v. Prompt the user to provide an answer to each multiplication problem. Each problem should appear on a new screen. Each screen should display your custom heading and the scoreboard as shown in Figure 3 (above). The scoreboard should update after each question is answers to reflect the number of questions asked and the number of questions answered correctly. User input must be a number only. If the user inputs a value other than a number, display a message and prompt the user to input a correct value.

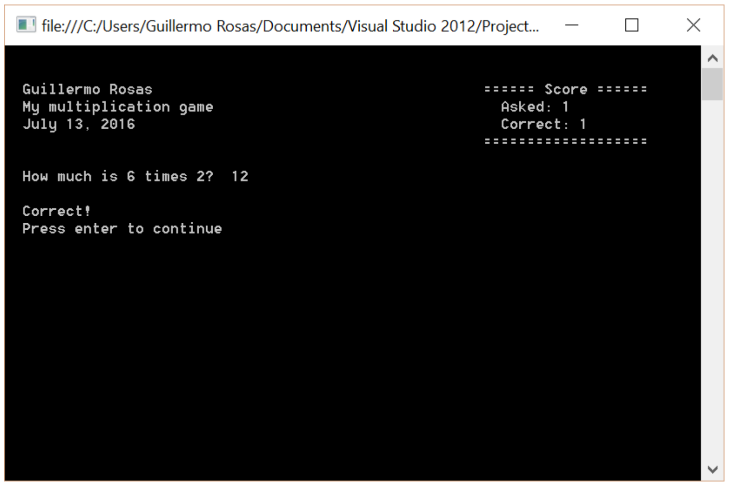
1. For each level of difficulty, use the random number generator to generate two numbers to multiply together:

a. For the “easy” gameplay, multiple single digits greater than zero together only (e.g., 1 – 9)

b. For the “medium” gameplay, each digit may be between 1 and 99.

c. For the “hard” gameplay, each digit may be between 1 and 1000.

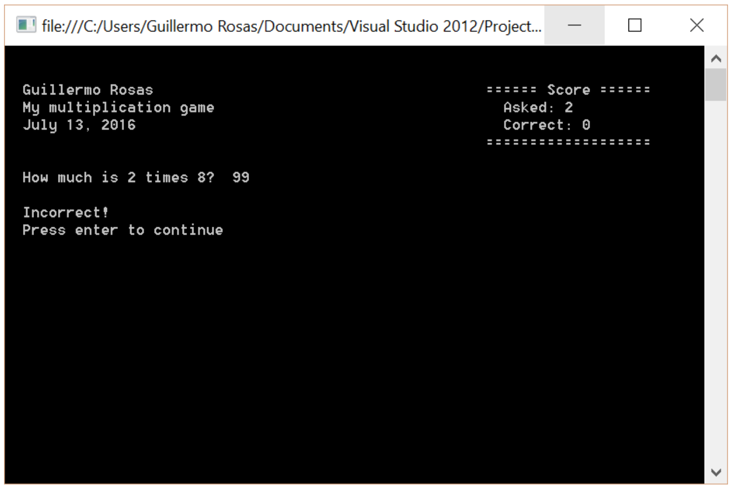
vi. Each time a user answers a problem, provide feedback and update the scoreboard as shown in Figure 4.


file:///C:/Users/Guillermo Rosas/Documents/Visual Studio 2012/Project...

```
Guillermo Rosas                                    ====== Score ======
My multiplication game                               Asked: 1
July 13, 2016                                        Correct: 1
                                                   ===================

How much is 6 times 2?  12

Correct!
Press enter to continue
```


file:///C:/Users/Guillermo Rosas/Documents/Visual Studio 2012/Project...

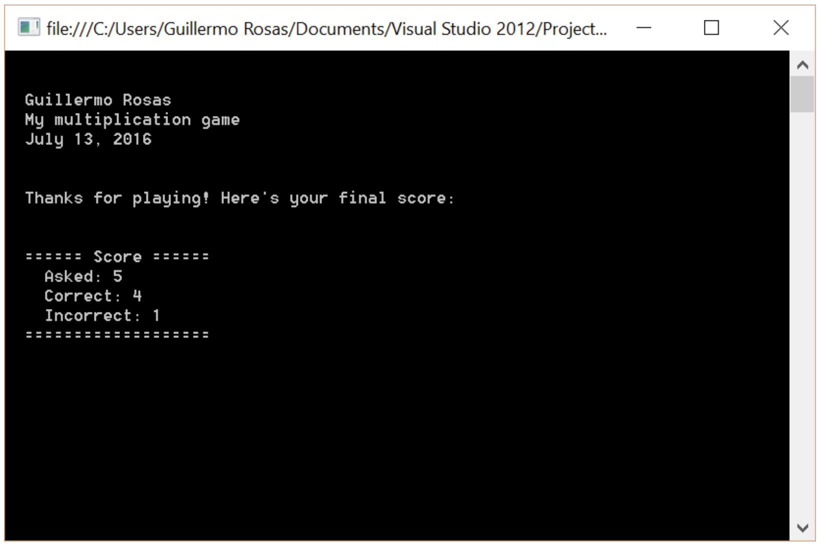
```
Guillermo Rosas                                    ====== Score ======
My multiplication game                               Asked: 2
July 13, 2016                                        Correct: 0
                                                   ===================

How much is 2 times 8?  99

Incorrect!
Press enter to continue
```

Figure 4

vii. The application should present the same number of problems the user chose to play in step 1aii. Once complete, display your custom heading with your application name. Make sure the scoreboard has been removed from the screen, then display a final score message. An example is shown below in Figure 5.


file:///C:/Users/Guillermo Rosas/Documents/Visual Studio 2012/Project...

```
Guillermo Rosas
My multiplication game
July 13, 2016


Thanks for playing! Here's your final score:


====== Score ======
  Asked: 5
  Correct: 4
  Incorrect: 1
===================
```

Figure 5

OTHER NOTES

- This assignment is a bit free form; however, you must use use an *enum* and a *switch* statement (sorry Todd) to determine how to generate numbers based on the chosen user level.
- Formatting is largely up to you; however, all output must be two spaces from the left, to align with the header.
- This is a bullet is just the requisite to make David Ream feel comfortable with the assignment.
- A sample application has been provided for you with the assignment. Reference the sample application to see how your application should work (note: there might be a bug in the score tracking – I'm writing this on the run – sorry! – but your solution should track the score correctly ;-)).
- This application was written on the run – there might be typos or conflicting instructions. If something doesn't make sense, email me ASAP – thanks!
- 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_6.zip** (where *LastName* is your own last name).
- Upload and submit the ZIP file to the Assignment 6 drop box in Blackboard.

Fin.