Name: **GuessANumberGUI**

Description: **Chapter 13 – Programming Exercise 7**

**Guess a random number game using radio buttons**

Write a C# GUI application that contains five Radio Buttons (numbered 1 to 5) that will be used in a guessing game. The application is to randomly choose one of the Radio Buttons as the winning Button. When the user clicks a Radio Button, display a message indicating whether or not the user’s selection (guess) matches the randomly generated number. Whenever the user guesses incorrectly, disable that Radio Button so it cannot be selected again. Include a Label that suggests there is a hidden hint for the user. When the mouse hovers over this Label, display another Label that states one particular Radio button is definitely not the randomly selected Radio Button. When the user guesses correctly, the game is over and the application must be restarted to play again. The exitButton is to be designated as the Cancel button and is to have an access key assigned to it. Design your GUI as shown.

Under the opening brace for public partial class Form1: Form – instantiate a Random object. Refer to this object using a name of your choice. Declare an integer variable to store the random number that will be generated in the Form1 Load event.

Create a Form1 Load event method for this application. Within the method, generate a random number between 1 and 5. Hide the exit button using (.Visible) equals false.

Create a Mouse Hover event method for the Label that suggests there is a hint available. Within the method, determine if the random number that was generated is 1. If it is, inform the user “*It’s not”* Radio Button 3. Otherwise, inform the user it is another Radio Button (random number – 1). Items in each display Label are to appear as shown.

Create a Mouse Leave event method for the Label that suggests there is a hint available. Within the method, clear the display label that informs the user which Radio Button it is not.

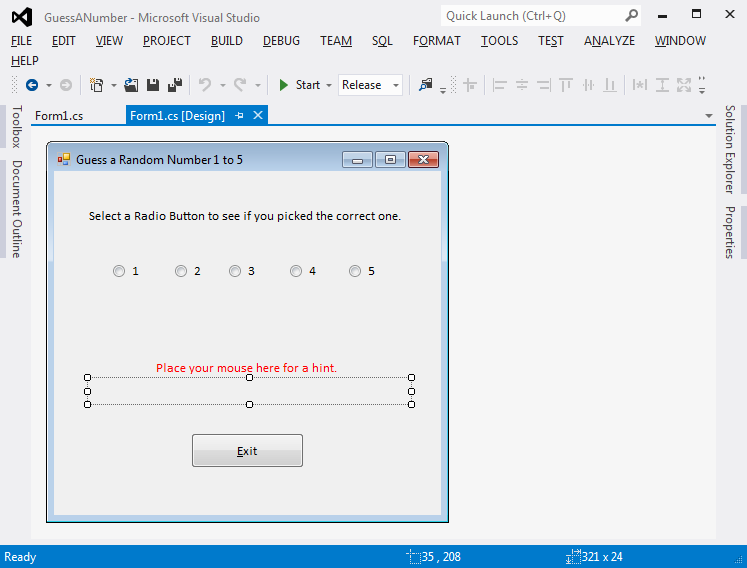
Create a Checked Changed event method for the Radio Button 1. Within the method, use an if statement to determine if Radio Button 1 was selected using (.Checked). If selected, use an if-else statement to determine if the random number is 1. If it is, display the appropriate message, make the exitButton visible using (.Visible) equals true, and set the focus to the exitButton. Otherwise, display the appropriate message and disable Radio Button 1 using (.Enabled) equals false. Create similar Checked Changed event methods for the remaining Radio Buttons.

Create a Click event method for the exitButton that terminates the application.

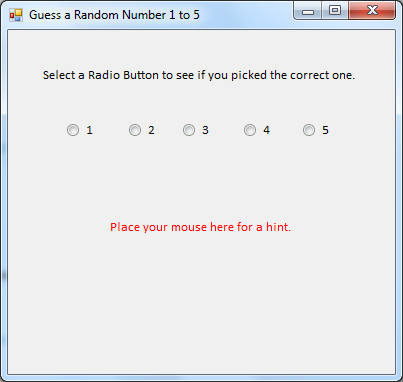
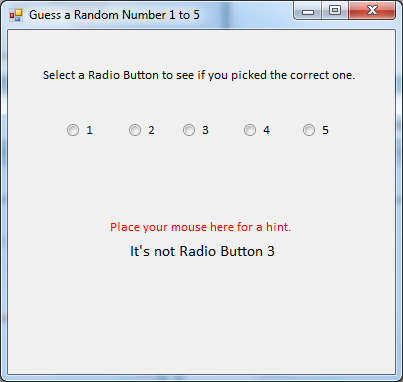
Complete the Pseudocode Template document for this programming request. A printed version is due upon arrival to class on lab day.

Use your completed Pseudocode document to create the C# application. Create and save the application to your classroom USB flash drive.

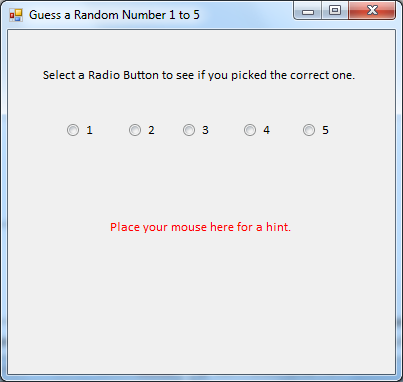
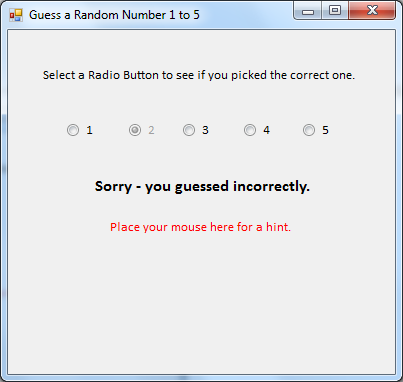
GUI in the Design Window



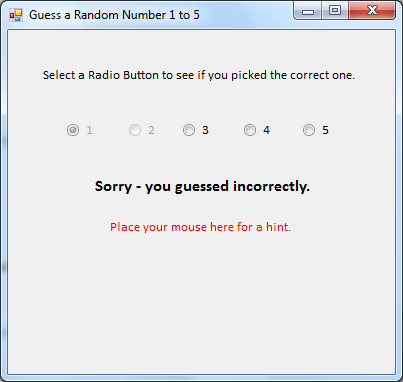
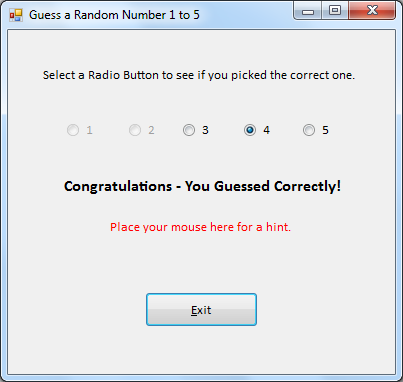
GUI When Started Sample Output (1) – MouseHover Hint Label

Sample Output (2) – MouseLeave Hint Label Sample Output (3) – Incorrect Guess

Sample Output (4) – Another Incorrect Guess Sample Output (5) – Correct Guess

*Notice the Radio Buttons for incorrect guesses are grayed-out meaning they are unavailable for selection.*

*Notice the “Exit” Button is only displayed after a correct guess.*