PROJECT: THE CONVERSATION

OVERVIEW

The focus of this coding activity it to allow the students to explore interacting with the user through the console. Formatting the screen and output to provide a consistent and quality experience for the user will be emphasized.

NOTES AND RESOURCES

- Students may use all resources available.
- Specifications and functionality requirements are separated into two categories, required and extended and will be assessed in the Skills module of Moodle
- **A solution will not be accepted without all required skills included.**

INSTRUCTIONS

1. Run the example application and view the example code to get a feel for the extent of the application.
2. Create a new solution (.NET C# Console) in Visual Studio and name it ***TheConversation***.
3. Develop a console application that has the following features, behaviors, and requirements. The purpose of the application is to have a brief "conversation" with the user, prompting them for input, and then responding to them based on their answers. . Run the program in **Debug Mode** frequently to test the code.
4. **Required Skills (refer to the example code)**
    a. Add an informational comment block at the top of the code block.

```
// ************************************************************
// Application:     The Conversation
// Author:          Velis, John E
// Description:     An application that has a conversation with the user
//                  about skiing.
// Date Created:    5/20/2016
// Date Revised:    7/7/2016
// ************************************************************
```

    b. Comment a variable section and declare a minimum of four string variables and one integer variables for user input. Choose appropriate variable names and use the **camel case** naming convention.

```
//
// variables
//
string userName;
string favoriteSki;
string typeOfSkiing;
string favoriteArea;
string userResponse;
```

c. Create and comment a minimum of three console screen sections; **Opening**, **User Interaction**, **Closing**. See Opening Screen example below.

```
//
//      **********************
//      *   Opening Screen   *
//      **********************
//
// set cursor invisible, background colors, foreground colors, and clear screen
//
Console.CursorVisible = false;
Console.BackgroundColor = openingClosingScreenBackground;
Console.ForegroundColor = openingClosingScreenForeground;
Console.Clear();


//
// display a opening screen
//
Console.WriteLine();
Console.WriteLine("\t\tThe Loan Consultant");
Console.WriteLine();
Console.WriteLine("\tPress any key to continue.");
Console.ReadKey();
```

d. Comment a user prompt section and write code to display a user prompt for each input and read each into the correct variable. Echo back the information provided by the user by displaying a message embedding the variable with the user information. (See the example below and note the three methods of embedding a variable into a message with the last three commented out.)

```
//
// get their name and echo it back
//
Console.WriteLine();
Console.Write("What is your name? ");
userName = Console.ReadLine();
Console.WriteLine("It is nice to meet you " + userName + ".");
//Console.WriteLine("It is nice to meet you {0}.", userName);
//Console.WriteLine($"It is nice to meet you {userName}.");
```

e. Prompt the user for values for each of the string variables and store them.

f. Prompt the user for values for each of the integer values, convert/parse the string value, and store them.

g. Explore of the following Console class properties to improve the User Interface (UI) and the User Experience (UE) making the application easier to use. The application should use colors, white space, and cleared screens to make the user experience flow more easily.

   1. BackgroundColor
   2. ForegroundColor
   3. CursorVisible
   4. WindowHeight
   5. WindowWidth
   6. Clear()
   7. Beep()

h. Create a visual theme for the application by changing the foreground and background colors.

i. Prompt the user with a yes/no question and use a decision statement block (**if / else**) making at least one message dependent on the user response to the prompt. (research C# conditionals)

j. Test and debug the application thoroughly.

k. Clean up the ***Program.cs*** file.

i. Delete all unnecessary code.
ii. Use tabbing consistently to denote all nesting. (Hint: use **Ctrl k + d**)
iii. Delete all unnecessary blank lines consistently only leaving blank lines to that assist in making the code more legible.

5. **Extended Skills (refer to the example code)**
    a. Include a double and a Boolean variable.
    b. Embed an integer or double variable value in a string and display it to the console.
    c. Use a decision statement block (**if / else if / else**) dependent on the user response to a prompt. (research C# conditionals)
    d. Use a nested decision statement block dependent on two user responses to prompts. (Research C# conditionals to complete this skill. (Research C# conditionals to complete this skill.)
    e. Display a message to the user integrating more than one of the string variable values.
    f. Display different messages based on the user's input of a string value.
    g. Display different messages based on the user's input of an integer, double or Boolean value.
    h. Perform a calculation based on user input and display the results.
    i. Change the color of a screen's foreground or background based on a user input value.
6. Test the application thoroughly and debug all issues.

SUBMIT THE ASSIGNMENT

1. Download and complete the Skills Checklist.
2. Submit to Moodle.
    a. Zip the solution folder into a single file.
    b. Click the ***M1 Project: The Conversation*** assignment link.
    c. Upload the zipped file.
    d. Submit the completed *Skills Checklist*.
    e. Click **Save Changes**.

PROJECT: THE CONVERSATION - SKILLS CHECKLIST

Author Joey Hoezee **Reviewer(s)** _____

Check all demonstrated skills and submit.

Required Skills	
Demonstrate creating a Visual Studio .NET C# Console solution and saving it to a local folder.	X
Demonstrate saving, zipping, and uploading a Visual Studio solution.	X
Demonstrate adding an information comment block.	X
Demonstrate adding useful and consistent commenting in a code file.	X
Demonstrate reading a string from the console and assigning it to a local variable.	X
Demonstrate embedding a string variable in a string and displaying to the console.	X
Demonstrate reading a string from the console, parsing or converting it, and assigning it to a local integer or double variable.	X
Demonstrate controlling the console using the ***Console*** class methods; **BackgroundColor, ForegroundColor, CursorVisible, WindowHeight, WindowWidth, Clear()**, and **Beep()**.	X
Demonstrate implementing an output decision based on a string variable.	X
Extended Skills	
Demonstrate embedding an integer or double variable in a string and displaying to the console.	X
Demonstrate implementing an output decision based on an integer or double variable.	
Demonstrate implementing an output decision based on a Boolean variable.	
Demonstrate implementing an **if/else if/.../else** code block.	X
Demonstrate implementing a nested decision code block.	
Perform a math operation on a minimum of two variables and save the result in another variable.	