In the following examples, **blue font** is user input.

1. Write a program that displays the following menu. Your program should not accept invalid number choices.

===============================

Rogue Character Creation Tool

===============================

Select from one of the following options:

1. Number of characters

2. Set character data

3. Write Characters data

4. Quit

Choice:

1. If the user provides an invalid choice, display an error message **in red** as seen below. The menu options will repeatedly be displayed until a valid choice is provided.

To change the font color:

Console.ForegroundColor = ConsoleColor.Red

To change the font color back to default:

Console.ResetColor()

===============================

Rogue Character Creation Tool

===============================

Select from one of the following options:

1. Number of characters

2. Set character data

3. Write Characters data

4. Quit

Choice: 5

Enter a valid menu option.

Select from one of the following options:

1. Number of characters

2. Set character data

3. Write Characters data

4. Quit

Choice:

1. Users should not be able to access option 2 and 3 unless they have first completed menu option 1.
2. Users should not be able to access menu option 3 unless they have first completed menu option 2.
3. If users attempt to access menu options prematurely, display an error message **in red** as seen below.

===============================

Rogue Character Creation Tool

===============================

Select from one of the following options:

1. Number of characters

2. Set character data

3. Write Characters data

4. Quit

Choice: 2

You must first enter the number of characters.

Select from one of the following options:

1. Number of characters

2. Set character data

3. Write Characters data

4. Quit

Choice: 3

You must first set character data.

Select from one of the following options:

1. Number of characters

2. Set character data

3. Write Characters data

4. Quit

Choice:

1. Menu option 1 asks the user to provide the number of characters they will be creating. Do not worry about handling bad input.
2. Menu option 2 asks the user to provide the character data and stores it within the character class. Example I/O is below.

==================

Character 1 Data

==================

Character Name: Bandit

Hit Points: 350

Attack Damage: 75

Attack Range: 15

Area of Effect: 5

==================

Character 2 Data

==================

Character Name: Thief

Hit Points: 200

Attack Damage: 65

Attack Range: 20

Area of Effect: 3

1. Menu option 3 writes all the character data to a text file within the current directory of the program.

A screenshot of a computer

Description automatically generated

1. Confirm the location of where the data was saved by displaying the file path to the user.
2. Your code must use object-oriented programming to store the character data.
   1. Create a “character” class with the following attributes:
      1. Name (string)
      2. Hit Points (integer)
      3. Attack Damage (integer)
   2. Create a “rogue” class that inherits from the character class with additional attributes:
      1. Attack Range (double)
      2. Area of Effect (double)
      3. Class Name (read only string)
3. Each menu option should have its own function or sub-procedure to execute the code.
4. The menu should continually display until the user decides to exit the program.