BREAKPOINT CODING PRACTICE

Working with Methods

Application Utilities

Pause Application (Void Method, No Parameters)

- Create a **PauseApplication** method with no return type or parameters
- Write a message to the Console asking the user to press a key
- Wait for the user to press a key before exiting the method

Exit Application (Void Method, No Parameters)

- Create an **ExitApplication** method with no return type or parameters
- Write a message to the Console asking the user to press a key
- Wait for the user to press a key, then exit the application

Display Colored Text (Void Method, Two Parameters)

- Create a **DisplayColoredText** method that has no return type but accepts to string parameters
    - String to hold the text to display
    - String to hold the color to change the Console.ForegroundColor
- Use a SWITCH statement to change the ForegroundColor to the desired color
- Write the text to the Console in the desired color
- Return the ForegroundColor to “gray.”

User Interaction

User Name (Return Data, No Parameters)

- Create a **GetUserName** method that has no parameters but returns a string value.
- Use the console to greet the user and ask for their name
- Store the name in a variable
- Return the name of the user

Days Until Birthday (Void Method, One Parameter)

- Create a **CalculateDaysUntilBirthday** method that has no return type but requires the name of the user.
    - You will not be changing the name of the user so don't use the REF keyword
- Ask the user for their birthday
- Display a message to the user informing them how many days they have until their next birthday
- Call the new method from your application

Does User Have Kids (Return Data, Two Parameters)

- Create a **DoesUserHaveKids** method what returns a Boolean value and requires the name of the user and an empty list of strings
    - You will not be changing the name of the user so don't use the REF keyword
    - A list is a reference type, so you will updating it in the method. You don't need the REF keyword.
- Ask the user if they have any kids
    - If they don't then return a value of FALSE
- If the user does have kids, have them populate the list with the names of their kids
- Return a value of TRUE

User Emotion Check (Void Method, One Parameter)

- Create a **CheckUserEmotions** method that has no return type but requires the name of the user.
- Ask the user how they are feeling
- Use a SWITCH statement to respond to a variety of emotions.
- Call the **DisplayColorText** method to respond to the user
    - Your response should be BLUE if the user is sad
    - Your response should be RED if the user is mad
    - Your response should be YELLOW if the user is happy.

Greet the User

User Interaction Method (Return Data No Parameters)

- Create a **GreetTheUser** method that has a string return type and requires no parameters.
- Create a variable to store the user name and call the **GetUserName** to populate the variable
- Call the **CalculateDaysUntilBirthday** passing in the user name
- Call the **PauseApplication** method
- Create a Boolean variable to store whether or not the user has kids
- Create an empty list of strings to store the names of the user's kids
- Call the **DoesUserHaveKids** method to populate the Boolean variable, pass in the user's name, and the empty list into the method.
- If the user has kids, loop through the list and with each of their kids a good life
- Call the **PauseApplication** method
- Call the **CheckUserEmotions** passing in the user name
- Return the user's name

Main Method

Perform the following actions in your Main method

- Create a variable to store the user's name
- Call the GreetUser method to run the application logic related to greeting the user
- Store the return of the GreetUser method in the variable
- Call the **PauseApplication** method
- …
- Call the **ExitApplication** method