ROJECT: FINCH CONTROL S4 (USER PROGRAMMING)

INSTRUCTIONS

1. Extend the application framework.
    a. Add the following **enum** just above the **class Program** declaration.

```
namespace Project_FinchControl
{
    /// <summary>
    /// User Commands
    /// </summary>
    31 references
    public enum Command
    {
        NONE,
        MOVEFORWARD,
        MOVEBACKWARD,
        STOPMOTORS,
        WAIT,
        TURNRIGHT,
        TURNLEFT,
        LEDON,
        LEDOFF,
        GETTEMPERATURE,
        DONE
    }
```

    b. **Method:** *void UserProgrammingDisplayMenuScreen(Finch finchRobot)*
        i. Use the same coding pattern as past menus.
        ii. Declare a tuple: *(int motorSpeed, int ledBrightness, double waitSeconds)* commandParameters
        iii. Declare a list of the enum Command**:** *List<Command> commands = new List<Command>()*
        iv. Display header.
        v. Display the menu and validate the user's response.
            a) Set Command Parameters
            b) Add Commands
            c) View Commands
            d) Execute Commands
            e) Return to Main Menu
        vi. Process user's choice using a switch/case block, calling the appropriate method.

c. **Method**: *(int motorSpeed, int ledBrightness, int waitSeconds) UserProgrammingDisplayGetCommandParameters()*
   i. Note: This method is returning a Tuple with three items.
   ii. Declare a tuple: *(int motorSpeed, int ledBrightness, double waitSeconds)* commandParameters
   iii. Display header.
   iv. Prompt, get, and validate the motor speed from the user.
   v. Prompt, get, and validate the LED brightness from the user.
   vi. Prompt, get, and validate the wait time from the user.
   vii. Echo the values provided by the user.
   viii. Return all of the values as a Tuple.
d. **Method:** *void UserProgrammingDisplayGetFinchCommands(List<Command> commands)*
   i. Display header.
   ii. Display instructions for user.
   iii. Add a while or do-while loop.
      1. Prompt the user for each new command.
      2. Parse, validate, and adds the new command to the commands list.
      3. Terminate when the user enters the "done" command.
   iv. Echo the user's commands (Hint: use a **foreach** loop).
   v. Call *DisplayContinuePrompt*.
e. **Method**: *void DisplayFinchCommands(List<Command> commands)*
   i. Display header.
   ii. Display all commands stored in the commands list.
   iii. Call *DisplayContinuePrompt*.
f. **Method**: *void DisplayExecuteFinchCommands(Finch finchRobot, List<Command> commands)*
   i. Display header.
   ii. Inform and prompt the user.
   iii. Execute all of the commands.
   iv. Display each command name as it is executed.
   v. Call *DisplayContinuePrompt*.

2. Test the application thoroughly.

SUBMIT THE ASSIGNMENT

1. Complete the **Skills Checklist.**
   a. [Face-Face only] Demonstrate the application to the instructor.
   b. [Online only] Upload the checklist in Moodle.
2. Push the VS solution to GitHub.
3. Submit to Moodle.
   a. Click the ***Project: Finch Control S4 (User Programming)*** assignment link.
   b. [Online only] Submit the completed **Skills Checklist**.
   c. [Online only] Submit a link to the streaming video walk-through.
   d. Submit the link to the GitHub repository with the solution.
   e. Click **Save Changes**.
4. After receiving a grade, refer to Moodle to review the graded rubric and additional comments.

PROJECT: FINCH CONTROL (USER PROGRAMMING) - SKILLS CHECKLIST

Author _____ **Reviewer(s)** _____

[In-class Students Only]
Code Share – Discuss the following during the Peer Review.

- Describe the flow of the application, walking through the application's major components.
- State one coding issue you encountered and how you resolved it.
- Highlight one unique block of code (method or function) that you developed and are particularly proud of. Share how the code block functions.
- State something that you learned during the development of this application that will be useful as you develop future applications.

[All Students]
Check all demonstrated skills and submit.

Skills	
Declare and initialize a list of simple data types.	V
Store user input into a list of enumerations.	V
Display a list of enumerations using a **foreach** loop.	V
Process a list of enumerations using a switch/case block.	V
Program the Finch robot with standard commands: light, sound, movement	V
Program the Finch robot with extended commands: combined light, sound, movement	V
Program the Finch robot with extended commands: sensor values	V
Store and process both command (enum) and command duration (int) as a list of tuple.	V
Validate user input with a feedback message: string value	
Validate user input with a feedback message: numeric value	
Validate user input with a feedback message: enumeration	