**Outline**

Define a class that encapsulates the implementation of an Arduino LED. Modify previous programs to leverage code reusability provided by implementing LED objects.

**Objectives**

* Use workplace and professional conventions (e.g., naming, indenting, commenting) correctly to write programs and internal documentation;
* Explain the importance of designing reusable code in computer programs;
* Explain fundamental object-oriented programming concepts (e.g., classes, objects, methods);
* Demonstrate the ability to create and use instance methods (e.g., constructors, mutators, accessors) in a computer program;
* Design a simple base class to represent objects or concepts in a problem statement, using program templates or skeletons;

**Prerequisites**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Prerequisite Module(s)** | **Level** | **Student Initial** | **Teacher Initial** | **Date** |
| Module B.1 Arduino Basic Blink | Level 3 |  |  |  |
| Module B.2 More Arduino Projects | Level 3 |  |  |  |

**Materials**

* Arduino Development Environment (IDE)
* Arduino proto board

**Level 0: Object Programming Documentation**

1. Locate on-line documentation for C++ object programming in the Arduino environment.
2. Explain the purpose and content of:
   1. Header (.h) files
   2. Class Code (.cpp) files
   3. Relationship between header and code files
3. Define and explain the following:
   1. Constructor
   2. Mutator Method
   3. Accessor Method
4. Explain how classes and objects are related but why they are very different.

**Level 1: Create LED Class**

1. Create a new project for this module and implement the header and code files for a cpp class that implements and encapsulated an Arduino LED.
2. The class should have the following methods:
   1. A constructor with a parameter for the port number
   2. Methods to turn on and turn off the LED
   3. A method to set the brightness of the LED
3. The class should have the following private class variables:
   1. Port number and a method to get the port number
   2. A Boolean variable for the on/off state of the LED and method to check if the LED is on or off
   3. An integer variable for the brightness of the LED and a method to return the brightness level
4. Locate on-line documentation for the proper naming and commenting of: Classes, Objects, Variables, and Methods. Make sure that your LED Class follows these naming conventions.

**Level 2: Basic LED Object**

1. Implement the “Fading” program from Module B2. using your LED class to validate and debug your work.

**Level 3: Extended LED Object**

1. Implement your “Extended Blinking” program from Module B.1 using your LED class to demonstrate code modularity and reusability.

**Level 4: Reusability Reflection**

1. Define “Code Reusability” and explain how your programs have improved through the use of the LED object.
2. Define “Code Encapsulation” and explain how your programs have improved through the use of the LED object.
3. Modify the brightness setting method in your LED class to make sure the parameter value is within the acceptable range of 0 to 255.
4. Explain how this code modification demonstrates the power of encapsulation and reusability.

**Achievement Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attainment Level** | **Student Initial** | **Teacher Initial** | **Date** |
| Level 0: Object Programming Documentation |  |  |  |
| Level 1: Create LED Class |  |  |  |
| Level 2: Basic LED Object |  |  |  |
| Level 3: Extended LED Object |  |  |  |
| Level 4: Reusability Reflection |  |  |  |