Objective

- To introduce Object-Oriented Programming (OOP)
- To write a class that is modeled using OOP paradigm

Introduction

An object is a way to bundle data and manipulate that data. This bundle is known as encapsulation. This is different from the procedural programming covered in the first part of the course, where data and code are separated. In the OOP model, a program is a collection of objects where these objects pass messages to each other. The goal of this lab is write simple classes using the OOP paradigm.

Background

Objected Oriented Programming (OOP):
Please refer to the OOP notes in the lecture for syntax.

Light Emitting Diode(LED):
We have used LEDs in the course before, but lets look at an LED from the OOP perspective. When connecting an LED to a microcontroller, the behaviors or actions we want the LED to have are:

1. Be able to connect to an I/O pin
2. Be able to turn on
3. Be able to turn off
4. Be able to inform its current state (i.e. on or off)
5. Be able to invert its state and report what it’s doing at the moment

Given the nature of OOP, we want to write a class that will contain the common features for any LED. Given the above actions for an LED, the common features or data that any LED object should keep track are:

1. I/O pin number
2. Current state of LED (i.e. ON or OFF state)

Note that in order to carry out the action of turning on and off an LED, we will still need to use the methods from CPU class. However, when we actually want to work with LED, we create a LED object, which will connect to a particular I/O and be in a particular state. Over time we can change the state of the LED or the pin to which it is connected. Note that doing something to one LED object will not affect other LEDs.
Materials

- Boe-Bot unit with Javelin Stamp
- Dr Java

Lab Instructions

Part I: Writing LED class

In the LED class, complete the code as stated in comments. Test your code by completing the main method in the FlashingLEDs class as stated in the comments.

Part II: Writing Player class

For this lab use Dr. Java. If you have not set up Dr. Java, please follow the following link:
http://www.seas.upenn.edu/~palsetia/java/installDrJava.html

Complete the lab below:
http://www.seas.upenn.edu/~ese112/spring10/java/Player/index.html

Post-Lab Questions

1. What is the use of a constructor in a class written using the OOP model?
2. What does the “new” operator do?
3. Explain the difference between global, local, and instance variables.
4. What are some areas where Object-Oriented Programming would be useful?
5. Consider the following code:

```java
public class Toy{
    int toyNum;
    public Toy(int num){
        int toyNum = num;
    }
    public int getToyNum(){
        return toyNum;
    }
} //end of Toy class
```

When the print statement executed in the main method from the TestToy class, we observe that value of n is 0, and not 5. Why does this happen? How will you fix it?

As always, submit all your Java programs (one per team) to Blackboard Digital Drop Box in one zipped folder using the format on the ESE112 website under the Course Information section.