**Lab 4: Interfaces and Parameters**

1. **ROS2 Custom Interfaces: (25 points)**
   1. You are going to implement a battery and LED panel. When the battery is empty, we power on an LED, and when the battery is full, we power it off.
      1. There is a battery node that is a client, a service called /set\_led, and an LED panel node that is a server, and a topic called /led\_panel\_state.
      2. The state of the battery is represented by a “battery\_state” variable inside the battery node, and the LED panel is represented by an integer array inside the LED panel node.
      3. At first, the battery is full, and all the LEDs are powered off ([0,0,0]). We will then “simulate” the battery state evolution: the battery is empty after 4 seconds, and then recharges to full after 6 seconds. It will constantly repeat this cycle.
      4. When the battery is empty, the battery node will send a request to the LED panel to power on one LED. Six seconds later, when the battery is full again, it will send another request to power off the LED. Your code can just continuously loop between these two states until you cress ctrl+c.
   2. You will create
      1. 1 node for the battery
      2. 1 node for the LED panel
      3. A custom msg definition for the /led\_panel\_state topic
      4. A custom srv for the /set\_led service

1. **ROS Parameters: (15 points)**
   1. Create a node that publishes a string on a topic: “Hi, my name is \_\_\_\_\_\_ and my favorite book is \_\_\_\_\_\_.” You will have to set the name and book parameters at run time by adding a name and book parameter.
   2. Go back to the LED panel node. You have an int array that represents the states of the LED, where 0 is off and 1 is powered on. Set this array with a parameter named “led\_states”.

An additional five points will be allocated in each homework assignment for style: Make sure your code is commented, neat, and variable names make sense. You should consult the Python style guide <https://peps.python.org/pep-0008/> as well as the ROS2 style guide: <https://docs.ros.org/en/rolling/The-ROS2-Project/Contributing/Code-Style-Language-Versions.html#python>.

**What to turn in**

* In a single zip file named DirectoryID\_Lab4.zip,
  + Your entire package that contains Q1 and Q2.
  + A video walkthrough showing you running Q1 and Q2.