**Getting Sensor Data**

Almost all Android devices come with built-in sensors. These sensors measure different aspects, such as motion, orientation, and environmental conditions like ambient light and temperature. These sensors are generally used to provide data to the users, as well as to the user’s applications with accuracy. This code lab taught me how to read data in using the Android sensor framework. This sensor framework is used to find available sensors on a device and retrieve the data from it. In this code lab, I learned how to implement sensor to an app for available sensors, retrieve specific sensors’ information, light and proximity levels. It is done by querying the sensor manager for those available sensors, register listeners for sensor data, and react to incoming sensor data.

The SensorManager class is a system service that allows you to access the device sensors. The Sensor class represents an individual sensor and defines constant for the available sensor types. Sensor.TYPE\_ALL constant indicates all the available sensors. Additionally, the sensor framework also provides the ability to register for and react for changes in sensor data. In this code lab app, it shows you how to get the sensors (light and proximity), listen for new sensor data, run the app and react to any changes made by the user. In order to be able to listen for new sensor data, we must implement from the SensorEventListener interface. You then must implement two of its methods: onSensorChanged() and onAccuracyChanged(). onSensorChanged() is called when there is new data available and onAccuracyChanged() is called if the sensor’s accuracy changes. During any data changes, the sensor framework generates a SensorEvent, in which the app must register listeners for these events and then handle these new sensor data in onSensorChanged() method. The app must also implement the onStart() and onStop() methods to register and unregister listeners.

Answer these questions

Question 1

Which of the following features are provided by the SensorManager class? (Choose all that apply)

* **Methods to register and unregister sensor listeners.**
* Methods to determine device orientation.
* Constants representing sensor types.
* Constants representing sensor accuracy.
* Methods to indicate whether a sensor is a wake-up sensor.

Question 2

In which Activity lifecycle method should you register your sensor listeners?

* onResume()
* onCreate()
* **onStart()**
* onRestart()

Question 3

What are best practices for using sensors in your app? (Choose all that apply)

* **Register listeners for only for the sensors you're interested in.**
* **Test to make sure that a sensor is available on the device before you use the sensor.**
* Check permissions for the sensor before you use it.
* Register a sensor listener for the slowest possible data rate.
* Don't block onSensorChanged() to filter or transform incoming data.