**Familiarity Review**

**Name:** Nkenta Uchechukwu Ferdinand

**Date:** October 16th, 2019

**Week:** 5

**Coding Topic:** **Threads, Executors and Runnable**

**Description of Understanding:** This week was more interesting thanks to “<https://www.journaldev.com/1020/thread-sleep-java>” I was able to get my hands on what it takes to create a Thread, Execute it and then make it Runnable. As far as **Thread** is concerned in Java, its feature allows concurrent execution between two or more part of the program. This is important as it helps to provide a well utilization of CPU. So, if we need to create a thread, we can:

* Extend the thread class.
* Implement a runnable interface.

**Executor:** This holds and manage the threads and ensures a runnable execution.

**Runnable:** Just as the name implies, this executes the code.

|  |  |
| --- | --- |
| **File** | **Git Link** |
| **Threads,**  **Executors & Runnable** | <https://github.com/nkenta/cit360/tree/master/threadsexecutorsrunnables> |

|  |  |
| --- | --- |
| **What should I be looking for?** | **Sandbox or Your code?** |
| Here there are two threads the primary and secondary thread. Each has an exception statement that helps control what is displayed when error occurs. In this simple program, you see where Thread.sleep is used to pause the thread for some seconds. As well as .start(); used to begin/start a thread. The For statements ensures that the program runs repeatedly until a satisfied condition is meet. | Mine |