CMSC 525 - Assignment #5
Test Plan Document
Gavin Alberghini

1. **What is the System Under Test (SUT)? Provide a link (if available) and a brief description.**
The SUT chosen for this assignment was named Operating-System-Simulation (OSS). OSS is a Java project consisting of several classes that simulate an operating system. This project was developed by me and is hosted in my personal github. Please utilize the links listed below for further information:

Repo Location -- https://github.com/gavinjalberghini/Operating-System-Simulation

Project Overview -- https://github.com/gavinjalberghini/Operating-System-Simulation/blob/master/README.md

Project Initial Requirements Elaboration -- https://github.com/gavinjalberghini/Operating-System-Simulation/blob/master/Documentation/CMSC%20312%20Project%202018.pdf

Project Documentation -- https://github.com/gavinjalberghini/Operating-System-Simulation/blob/master/Documentation/Operating%20System%20Documentation.pdf

2. **Will you test every class in the SUT? If not, which classes will be tested?**
For this assignment, eight of the twenty existing classes will be tested. These numbers include nested class definitions. There are six .java files from a total of sixteen that undergo analysis for this assignment. The selected files and corresponding test files are listed below:

1. OS.Frame – test.OS.FrameTest
2. OS.MMU – test.OS.MMUTest
3. OS.Memory – test.OS.MemoryTest
4. OS.PCB – test.OS.PCBTest
5. OS.PDT – test.OS.PDTTest
6. OS.Process – test.OS.ProcessTest

3. **Will you do manual, automated testing (fuzzing), or both?**
For this assignment, all tests have been implemented manually or in a slightly automated fashion using Junit skeleton generation via JUnitGenerator V2.0 (an IntelliJ plugin).

4. **What coverage criteria will be used?**
For this assignment the following coverage criteria are computed:

1. Line Coverage
2. Method Coverage
3. Class Coverage
4. Strong Mutation Coverage

CMSC 525 - Assignment #5
Test Plan Document
Gavin Alberghini

5. **Will you be using any external tools to compute coverage or perform other tasks (outside of Junit/Java/Python)?**
The tools utilized are listed below with brief descriptions:

1. JUnitGenerator V2.0 Plugin – This tool allows for automatic skeleton generation of Junit tests based off of a supplied class. This tool can also be configure to utilize user defined templates for more tailored generation. This tool can be found here: https://plugins.jetbrains.com/plugin/3064-junitgenerator-v2-0.

2. IntelliJ IDEA Coverage Plugin – This tool reports line, method, and class coverage based on a set of classes given the test files meant to cover them. Descriptions of this tool can be found here: https://www.jetbrains.com/help/idea/code-coverage.html

3. PIT Mutation Testing Plugin – This tool is utilized to measure mutation coverage. For this assignment, it was utilized for generating strong mutation coverage. This tool can be found here: https://pitest.org