UNIVERSITY OF VIRGINIA
DATA SCIENCE INSTITUTE

CS 5012: Foundations of Computer Science

Group Activity: Concurrency

Directions: Prior to the live session, be sure to complete Activity Example 1 and Example 2. You will be working in small groups to run a series of python files and answer questions about the output.

First, organize yourselves into small groups of 3-4 through Piazza. You will be working on these files in the live session through breakout groups, so time is of the essence. Here are a few recommendations for organizing yourselves within your group.

1. Load the document, Group Activity: Concurrency, into Google Drive and ensure all group members have editing access.
2. Designate roles for each member of the group. Here are some suggested roles:
    a. **Recorder:** This person will be responsible for documenting the team's answers to the questions in Group Activity: Concurrency
    b. **Code Runners (2):** These people will be responsible for running the python code. One of these people will be responsible for sharing their screen in the breakout room. The other code runner will serve as another set of eyes to ensure the output and explanation match what the other code runner is displaying.

You will spend approximately 10 minutes on each of the python files. Follow the instructions in the file for each file and discuss amongst yourselves. During the last 10 minutes of the live session, you will convene as a whole group and discuss any surprises in the output and parts of code that you are having difficulty with. You may not be able to finish the activity within the live session, but it will be our goal to do so. By the end of the week, you will need to submit your file, Group Activity: Threads, for grading.

Submission Instructions

1. Include group members names and UVA computing IDs at the top of your document

Example 1

(10 minutes) Python file **example1.py** [**introduction to threads; creating threads**].

1. Before running the Python file, examine each of the lines of code in the file. Predict the output when this file is run.
2. Still, prior to running the Python file, comment/discuss on the parts of the code you understand well, and parts of the code you least understand.
3. Run the Python file and observe the output.
4. Examine the output and explain what you see in your own words.
5. Did the output match what you predicted? Explain.

Example 2

(10 minutes) Python file **example2.py** [**introduction to threads; counting words**].

1. Before running the Python file, examine each of the lines of code in the file. Predict the output when this file is run.
2. Still, prior to running the Python file, comment/discuss on the parts of the code you understand well, and parts of the code you least understand.
3. Run the Python file and observe the output.
4. Examine the output and explain what you see in your own words.
5. Did the output match what you predicted? Explain.

Example 3

(10 minutes) Python file **example3.py** [**race conditions**].

1. Before running the Python file, examine each of the lines of code in the file. Predict the output when this file is run.
2. Still, prior to running the Python file, comment/discuss on the parts of the code you understand well, and parts of the code you least understand.
3. Run the Python file and observe the output.
4. Examine the output and explain what you see in your own words.
5. Did the output match what you predicted? Explain.

Example 4

(10 minutes) Python file **example4.py** [**synchronizing object access; locks**].

1. Before running the Python file, examine each of the lines of code in the file. Predict the output when this file is run.
2. Still, prior to running the Python file, comment/discuss on the parts of the code you understand well, and parts of the code you least understand.
3. Run the Python file and observe the output.
4. Examine the output and explain what you see in your own words.
5. Did the output match what you predicted? Explain.

Example 5

(10 minutes) Python file **example5.py** [**avoiding deadlocks; conditions**].

1. Before running the Python file, examine each of the lines of code in the file. Predict the output when this file is run.
2. Still, prior to running the Python file, comment/discuss on the parts of the code you understand well, and parts of the code you least understand.
3. Run the Python file and observe the output.
4. Examine the output and explain what you see in your own words.
5. Did the output match what you predicted? Explain.