Guidelines on running the Java program

\* To run the Java program, ensure that you have the latest JDK installed on your PC.

\* Go to the command prompt, and use the command cd to take you to where the java file is. For example, if it is in the Downloads folder, you should use the following command: cd Downloads

\* Compile your Java file by using the following command prompt: Javac Main.java

\* After successful compilation, use the following command to run the program: java Main

\* Input any number of your choice

Additional Information

\* I have also written a C program but I am not sure if it works.

\* Kindly run the C file first and test it out. If it is correct and meets the requirements, you may use it.

**General Grading Guidelines:**The following deductions will apply to all programming assignments.  
  
1. Any program that does not compile will result in a **zero** for the assignment. No exceptions will be made for ‘accidental’ uploads, so check your code before submitting.  
2. Poor Formatting and programming style will result in a **15% deduction**. Please see the style guidelines to avoid this deduction.  
3. The code will go through 2 steps of plagiarism detection. Zero tolerance for academic misconduct!  If you commit any form of academic misconduct in this course, you WILL be reported to the FIU Student Conduct and Academic Affairs office.

**Programming Style Guidelines:**  
  
The major purpose of programming style guidelines is to make programs easy to read and understand. Good programming style helps make it possible for a person knowledgeable in the application area to quickly read a program and understand how it works.  
  
1. Your program should begin with a comment that briefly summarizes what it does. For this course, this comment should also include your Name and pantherID.  
2. Use additional comments when needed in order for a reader to understand what is happening (see also, point 3).  
3. Variable names and function names should be sufficiently descriptive so that a knowledgeable reader can easily understand what the variable means and what the function does. If this is not possible, comments should be added to make the meaning clear.  
4. Use consistent indentation to emphasize block structure.  
5. Use names of moderate length for variables. Most names should be between 2 and 12 letters long.  
6. Use either underscores or capitalization (camelNaming) for compound names for variables. e.g: tot\_vol, total\_volumn, or totalVolumn.

Objective

Implement FCFS and SJF scheduling algorithms in C and calculate the average waiting time and average turnaround time of concurrent processes.

Description:

* Input:  the number of processes along with the burst time and arrival time for each process. In addition to the desired scheduling algorithm (FCFS or SJF). The program should expect a file name and a scheduling algorithm as arguments.

      Ex: myprogram myfile.txt FCFS

* The file should contain the number of processes in the first line. Then, each following line should contain the burst time and arrival time of one process as follows:

```

<number of processes>

<process 1 arrival time> <process 1 burst time>

<process 2 arrival time> <process 2 burst time>

…

```

* You can find an example of the input file [here](https://fiu.instructure.com/courses/189076/files/30157724?wrap=1)[Download here](https://fiu.instructure.com/courses/189076/files/30157724/download?download_frd=1)and the code for reading the content of the file [here](https://fiu.instructure.com/courses/189076/files/30157728?wrap=1)[Download here](https://fiu.instructure.com/courses/189076/files/30157728/download?download_frd=1). Note that this example is NOT the test case for grading.
* the program should calculate the waiting time and turnaround time for each process if the selected scheduling algorithm was used and output the average waiting time and average turnaround time.
* Output: order of execution of the processes (ex: P1 -> P3 -> P6) and the average waiting time and average turnaround time.

**Sample Output:**

*Order of Execution: P1 -> P2 -> P3*  
*Average Waiting Time: 2.33*  
*Average Turnaround Time: 8.33*

**Submission Guidelines:**

Submit the C source code along with a brief report explaining the design choices and the lessons learned during the implementation. Discuss any challenges encountered and how they were addressed.

Remember to include the following as a comment at the beginning of your program:

* Your name
* Your panther ID
* A description of the program

Your code will be tested on a Linux machine with gcc compiler

**Deliverables:**

your **C source code**(e.g. *myfirstlastname*.c ) and **PDf**report.

PLEASE SAVE AS THIS

Kenneth Richards

Panther ID - 1047305