CSE4510: Operating Systems Laboratory (Fall’23)
Offline-4 sec: C Total-20 marks

Submission Guideline:

- Put all your java files and output text file inside a folder
- Rename output text files as studentID_output_FCFS.txt and studentID_output_ISJF.txt
- Rename the folder according to your student ID
- ZIP the folder and upload it. The name of the zipped folder should also be your student ID

- *Do not copy codes from the internet. You will be given a template code. You must use that and complete those.*
- *Do not collect codes from previous trimesters’ students. Such copies are easily detectable and will be penalized with **-100%.***

Simulate two CPU scheduling algorithms.

- **[10 marks]** FCFS [Code template provided and explained in the class]
- **[10 marks]** Impatient-SJF [Make necessary changes in FCFS to implement this]

You must use the template provided in the class.

In FCFS, when the CPU becomes free, pick the process from the ready queue that arrived earlier than all the other ready processes.

In Impatient-SJF, when the CPU becomes free, pick the process from the ready queue that has the shortest burst time. But this time, no process will wait for more than 5 time units in the queue. So if the process has already waited for more than 5 time units, discard the process without running on CPU. Otherwise, run the process as usual.

Required outputs for each algorithm:

- During the simulation (run for 35 time units):
  - If any process is running, print the process id and global time at each timestamp.
  - If no process is running, print the global time and mention that no process is ready.
    (See the output provided in slide for better understanding)

- At the end of the simulation:
  - Print the minimum waiting time and maximum waiting time and their corresponding process IDs.
  - Print average waiting time and average turnaround time.

- For this task, you can implement additional variables/methods in any class.

Prepare a text file (output.txt) with your output lines. You can do this by simply copying the console output to a text file. Otherwise, you can use Java File IO to do it. Follow this link for reference: Java Create and Write To Files. Submit the text file along with your source codes.
In time of evaluation, you also have to generate an output file. Therefore, write code such that your program can generate output file for different inputs.