Assignment 1

Process Scheduling in Operating Systems

(Java Implementation)

Objective

Implement and compare different CPU scheduling algorithms by calculating:

- **Average Waiting Time**
- **Average Turnaround Time**

All programs must receive **user input** (number of processes, burst time, arrival time if required, priority if required).

Algorithms to Implement

You must write a Java program (single program or separate classes) that simulates the following CPU scheduling algorithms:

1. **First-Come First-Served (FCFS)** — Non-preemptive
2. **Shortest Job First (SJF)** — Non-preemptive
3. **Shortest Remaining Time First (SRTF)** — Preemptive
4. **Priority Scheduling** — Non-preemptive (lower number = higher priority unless stated otherwise)

Expected Output for Each Algorithm

For each process:

- Waiting Time
- Turnaround Time

And at the end:

- **Average Waiting Time**

- **Average Turnaround Time**

Detailed Instructions

i. First Come First Served (FCFS)

1. Read number of processes.
2. Read burst time (and arrival time).
3. Sort processes by arrival time.
4. Compute waiting time:
   - `WT[0] = 0`
   - `WT[i] = WT[i-1] + BT[i-1]`
5. Compute turnaround time:
   - `TAT[i] = WT[i] + BT[i]`
6. Display all results + averages.

---

ii. Shortest Job First (SJF) – Non-preemptive

1. Read number of processes and burst time.
2. Sort processes by burst time (arrival time ignored).
3. Compute waiting times exactly like FCFS but with sorted order.
4. Compute turnaround times.
5. Display all results + averages.

---

iii. Shortest Remaining Time First (SRTF) – Preemptive SJF

1. Read number of processes, burst time **and arrival time**.
2. For each time unit until all processes complete:
   - Select process with **minimum remaining time** among arrived processes.
   - Decrease its remaining time by 1.
   - When remaining time hits 0:
     - Record completion time.
3. After all processes complete:
   - `TAT = completion − arrival`
   - `WT = TAT − burst`
4. Display results + averages.

---

iv. Priority Scheduling (Non-preemptive)

1. Read burst time, priority (arrival time optional).
2. Sort processes by priority.
3. Compute waiting times and turnaround times.
4. Display all results + averages.

Assignment Question

Question:

Write a Java program to simulate the following **CPU scheduling algorithms** and compute **average waiting time** and **average turnaround time**.
Your program must take user input for all required fields.

a) **FCFS**
b) **SJF**

c) **preemptive SRTF**
c) **Priority Scheduling**

Sample Input:

Process	Arrival Time	Burst Time	Priority
P0	0	4	2
P1	1	3	3
P2	2	1	4
P3	3	5	5
P4	4	2	5