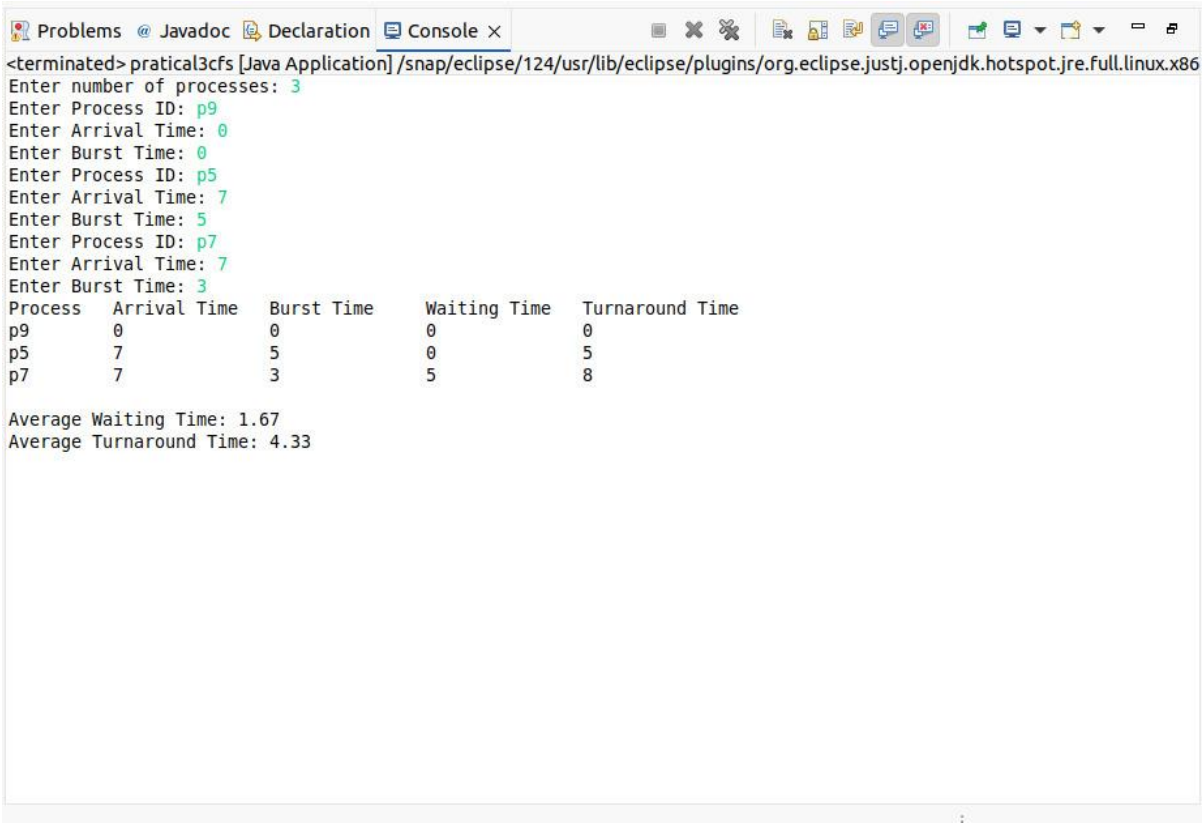
```java
package fcfs;

import java.util.*;

/**
 * Represents a process with necessary scheduling attributes.
 */
class Process {
    private String processId;
    private int arrivalTime;
    private int burstTime;
    private int waitingTime;
    private int turnaroundTime;

    public Process(String processId, int arrivalTime, int burstTime) {
        this.processId = processId;
        this.arrivalTime = arrivalTime;
        this.burstTime = burstTime;
    }

    // Getters
    public String getProcessId() { return processId; }
    public int getArrivalTime() { return arrivalTime; }
    public int getBurstTime() { return burstTime; }
    public int getWaitingTime() { return waitingTime; }
    public int getTurnaroundTime() { return turnaroundTime; }

    // Setters
    public void setWaitingTime(int waitingTime) { this.waitingTime = waitingTime; }
    public void setTurnaroundTime(int turnaroundTime) { this.turnaroundTime = turnaroundTime; }
}

/**
 * Scheduler class implementing FCFS algorithm logic.
 */
class Scheduler {

    public void calculateFCFS(List<Process> processes) {
        // Sort processes by arrival time
        processes.sort(Comparator.comparingInt(Process::getArrivalTime));

        int currentTime = 0;

        for (Process p : processes) {
            if (currentTime < p.getArrivalTime()) {
                currentTime = p.getArrivalTime();  // CPU idle
            }

            int waitingTime = currentTime - p.getArrivalTime();
            int turnaroundTime = waitingTime + p.getBurstTime();

            p.setWaitingTime(waitingTime);
            p.setTurnaroundTime(turnaroundTime);

```

```
            currentTime += p.getBurstTime();
        }
    }

    public void display(List<Process> processes) {
        System.out.printf("%-10s%-15s%-15s%-15s%-15s%n",
                "Process", "Arrival Time", "Burst Time", "Waiting Time", "Turnaround Time");

        for (Process p : processes) {
            System.out.printf("%-10s%-15d%-15d%-15d%-15d%n",
                    p.getProcessId(), p.getArrivalTime(), p.getBurstTime(),
                    p.getWaitingTime(), p.getTurnaroundTime());
        }

        double avgWT = processes.stream().mapToInt(Process::getWaitingTime).average().orElse(0);
        double avgTAT = processes.stream().mapToInt(Process::getTurnaroundTime).average().orElse(0);

        System.out.printf("%nAverage Waiting Time: %.2f%n", avgWT);
        System.out.printf("Average Turnaround Time: %.2f%n", avgTAT);
    }
}

/**
 * Main class to run the FCFS scheduler.
 */
public class pratical3cfs {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        List<Process> processList = new ArrayList<>();

        System.out.print("Enter number of processes: ");
        int n = scanner.nextInt();

        for (int i = 0; i < n; i++) {
            System.out.print("Enter Process ID: ");
            String pid = scanner.next();

            System.out.print("Enter Arrival Time: ");
            int at = scanner.nextInt();

            System.out.print("Enter Burst Time: ");
            int bt = scanner.nextInt();

            processList.add(new Process(pid, at, bt));
        }

        Scheduler scheduler = new Scheduler();
        scheduler.calculateFCFS(processList);
        scheduler.display(processList);
    }
}
```

Output :


Problems
Javadoc
Declaration
Console
<terminated> pratical3cfs [Java Application] /snap/eclipse/124/usr/lib/eclipse/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.linux.x86
Enter number of processes: 3
Enter Process ID: p9
Enter Arrival Time: 0
Enter Burst Time: 0
Enter Process ID: p5
Enter Arrival Time: 7
Enter Burst Time: 5
Enter Process ID: p7
Enter Arrival Time: 7
Enter Burst Time: 3
Process Arrival Time Burst Time Waiting Time Turnaround Time
p9 0 0 0 0
p5 7 5 0 5
p7 7 3 5 8
Average Waiting Time: 1.67
Average Turnaround Time: 4.33