```
package ass5FCFS;

import java.util.Scanner;
public class ass5FCFS {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n, temp;
        float avgtat = 0, avgwt = 0;
        System.out.println("*** First Come First Serve Scheduling ***");
        System.out.print("Enter Number of Process: ");
        n = sc.nextInt();
        int process[] = new int[n];
        int arrivaltime[] = new int[n];
        int burstTime[] = new int[n];
        int completionTime[] = new int[n];
        int TAT[] = new int[n];
        int waitingTime[] = new int[n];
        for (int i = 0; i < n; i++) {
            process[i] = (i + 1);
            System.out.print("\nEnter Arrival Time for processor " + (i + 1) + ":");
            arrivaltime[i] = sc.nextInt();
            System.out.print("Enter Burst Time for processor " + (i + 1) + ": ");
            burstTime[i] = sc.nextInt();
        }
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (arrivaltime[i] > arrivaltime[j]) {
                    temp = process[j];
                    process[j] = process[i];
                    process[i] = temp;
                    temp = arrivaltime[j];
                    arrivaltime[j] = arrivaltime[i];
                    arrivaltime[i] = temp;
                    temp = burstTime[j];
                    burstTime[j] = burstTime[i];
                    burstTime[i] = temp;
                }
            }
        }
        for (int i = 0; i < n; i++) {
            if (i == 0) {
                completionTime[i] = arrivaltime[i] + burstTime[i];
            } else {
                if (arrivaltime[i] > completionTime[i - 1]) {
                    completionTime[i] = arrivaltime[i] + burstTime[i];
                } else {
                    completionTime[i] = completionTime[i - 1] + burstTime[i];
                }
            }
        }

        System.out.println("\n*** First Come First Serve Scheduling ***");
        System.out.println("Processor\tArrival time\tBrust time\tCompletion Time\t\tTurn around time\tWaitin
g time");
        System.out.println(
                "---------------------------------------------------------------------------------------------------------");
        for (int i = 0; i < n; i++) {

            TAT[i] = completionTime[i] - arrivaltime[i];
            waitingTime[i] = TAT[i] - burstTime[i];
            avgtat += TAT[i];
            avgwt += waitingTime[i];
            System.out.println("P" + process[i] + "\t\t" + arrivaltime[i] + "ms\t\t" + burstTime[i] + "ms\t\t"
                    + completionTime[i] + "ms\t\t\t" + TAT[i] + "ms\t\t\t" + waitingTime[i] + "ms");
        }
        System.out.println("\nAverage turn around time of processor: " + (avgtat / n)
                + "ms\nAverage waiting time of processor: " + (avgwt / n) + "ms");
        sc.close();
    }
}

/*
 * *** First Come First Serve Scheduling ***
Enter Number of Process: 4

Enter Arrival Time for processor 1:0
Enter Burst Time for processor 1: 2

Enter Arrival Time for processor 2:1
Enter Burst Time for processor 2: 2

Enter Arrival Time for processor 3:5
Enter Burst Time for processor 3: 3

Enter Arrival Time for processor 4:6
Enter Burst Time for processor 4: 4

*** First Come First Serve Scheduling ***
Processor Arrival time Brust time Completion Time  Turn around time Waiting time
---------------------------------------------------------------------------------------------------------
P1  0ms  2ms  2ms   2ms   0ms
P2  1ms  2ms  4ms   3ms   1ms
P3  5ms  3ms  8ms   3ms   0ms
P4  6ms  4ms  12ms   6ms   2ms

Average turn around time of processor: 3.5ms
Average waiting time of processor: 0.75ms
Gantt chart = p1 p2//==// p3 p4
           0 2 4     5   8  12

criteria = "Arrival time"
Mode = non - preemitive
turn around time = CT - AT
waiting time = TAT-BT
RT = Response Time = Time at which a process go the CPU cost - arrival time
*/
```