```
package ass5NONSJF;

import java.util.Scanner;

public class ass5NONSJF {
  public static void main(String args[]) {
    Scanner sc = new Scanner(System.in);
    System.out.println("*** Shortest Job First Scheduling (Non Preemptive) ***");
    System.out.print("Enter no of process:");
    int n = sc.nextInt();
    int process[] = new int[n];
    int arrivaltime[] = new int[n + 1];
    int burstTime[] = new int[n + 1];
    int completionTime[] = new int[n];
    int TAT[] = new int[n];
    int waitingTime[] = new int[n];
    int temp, k = 1, time = 0;
    int min = 0, sum = 0, compTotal = 0;
    float avgwt = 0, avgTAT = 0;

    for (int i = 0; i < n; i++) {
      System.out.println(" ");
      process[i] = (i + 1);
      System.out.print("Enter 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; i++) {
      for (int j = 0; 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 j = 0; j < n; j++) {
      time = time + burstTime[j];
      min = burstTime[k];
      for (int i = k; i < n; i++) {
        if (time >= arrivaltime[i] && burstTime[i] < min) {
          temp = process[k];
          process[k] = process[i];
          process[i] = temp;
          temp = arrivaltime[k];
          arrivaltime[k] = arrivaltime[i];
          arrivaltime[i] = temp;
          temp = burstTime[k];
          burstTime[k] = burstTime[i];
          burstTime[i] = temp;
        }
      }
      k++;
    }
    waitingTime[0] = 0;
    for (int i = 1; i < n; i++) {
      sum = sum + burstTime[i - 1];
      waitingTime[i] = sum - arrivaltime[i];
      avgwt += waitingTime[i];
    }
    for (int i = 0; i < n; i++) {
      compTotal = compTotal + burstTime[i];
      completionTime[i] = compTotal;
      TAT[i] = compTotal - arrivaltime[i];
      avgTAT += TAT[i];
    }

    System.out.println("\n*** Shortest Job First Scheduling (Non Preemptive) ***");
    System.out.println("Processor\tArrival time\tBrust time\tCompletion Time\t\tTurn around time\tWaiting time");
    System.out.println(
        "-----------------------------------------------------------------------------------------------------------------------------");
    for (int i = 0; i < n; 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");
    }
    avgTAT /= n;
    avgwt /= n;
    System.out.println("\nAverage turn around time is " + avgTAT);
    System.out.println("Average waiting time is " + avgwt);
    sc.close();
  }
}

/**** Shortest Job First Scheduling (Non Preemptive) ***
Enter no of process:5

Enter Arrival Time for processor 1:2
Enter Burst Time for processor 1: 6

Enter Arrival Time for processor 2:5
Enter Burst Time for processor 2: 2

Enter Arrival Time for processor 3:1
Enter Burst Time for processor 3: 8

Enter Arrival Time for processor 4:0
Enter Burst Time for processor 4: 3

Enter Arrival Time for processor 5:4
Enter Burst Time for processor 5: 4

*** Shortest Job First Scheduling (Non Preemptive) ***
```

```
Processor Arrival time Brust time Completion Time  Turn around time Waiting time
------------------------------------------------------------------------------------------------------------
P4  0ms  3ms  3ms   3ms   0ms
P1  2ms  6ms  9ms   7ms   1ms
P2  5ms  2ms  11ms   6ms   4ms
P5  4ms  4ms  15ms   11ms   7ms
P3  1ms  8ms  23ms   22ms   14ms

Gannat chart : p4  p1  p2  p5  p3
              0  3   9   11  15   23
criteria = "Burst Time"
Mode = "Non preemptuve"
TAT = CT - AT
WT = TAT - BT
Average turn around time is 9.8
Average waiting time is 5.2
 * */
```