Shortest Job First Scheduling (Non Preemptive):

```
import java.util.Scanner;

public class NonSJF {

  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();
  }
}
```

Output:

```
*** Shortest Job First Scheduling (Non Preemptive) ***
Enter no of process:4


Enter Arrival Time for processor 1:1
Enter Burst Time for processor 1: 2
Enter Arrival Time for processor 2:1
Enter Burst Time for processor 2: 4

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

Enter Arrival Time for processor 4:2
Enter Burst Time for processor 4: 4

*** Shortest Job First Scheduling (Non Preemptive) ***
Processor    Arrival time   Brust time   Completion Time     Turn around time     Waiting time
-----------------------------------------------------------------------------------------------------
   P2           1ms           4ms            4ms                  3ms                0ms
   P1           1ms           2ms            6ms                  5ms                3ms
   P4           2ms           4ms            10ms                 8ms                4ms
   P3           3ms           4ms            14ms                 11ms                7ms

Average turn around time is 6.75
Average waiting time is 3.5
```