Non Priority Sheduling:

```
import java.util.Scanner;

class NonPriorityScheduling {

  public static void main(String[] args) {

    System.out.println("*** Priority Scheduling (Non Preemptive) ***");

    System.out.print("Enter Number of Process: ");
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    int process[] = new int[n];
    int arrivaltime[] = new int[n];
    int burstTime[] = new int[n];
    int completionTime[] = new int[n];
    int priority[] = new int[n];
    int TAT[] = new int[n];
    int waitingTime[] = new int[n];
    int arrivaltimecopy[] = new int[n];
    int burstTimecopy[] = new int[n];
    int max = -1, min = 9999;
    int totalTime = 0, tLap, temp;
    int minIndex = 0, currentIndex = 0;
    double avgWT = 0, avgTAT = 0;
    for (int i = 0; i < n; i++) {
      process[i] = (i + 1);
      System.out.println("");
      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();
```

```
  System.out.print("Enter Priority for " + (i + 1) + " process: ");
  priority[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[i];
    process[i] = process[j];
    process[j] = temp;
    temp = arrivaltime[j];
    arrivaltime[j] = arrivaltime[i];
    arrivaltime[i] = temp;
    temp = priority[j];
    priority[j] = priority[i];
    priority[i] = temp;
    temp = burstTime[j];
    burstTime[j] = burstTime[i];
    burstTime[i] = temp;
   } else if (arrivaltime[i] == arrivaltime[j] && priority[j] > priority[i]) {
    temp = process[i];
    process[i] = process[j];
    process[j] = temp;
    temp = arrivaltime[j];
    arrivaltime[j] = arrivaltime[i];
    arrivaltime[i] = temp;
    temp = priority[j];
    priority[j] = priority[i];
    priority[i] = temp;
    temp = burstTime[j];
    burstTime[j] = burstTime[i];
    burstTime[i] = temp;
   }
```

```
  }
}
System.arraycopy(arrivaltime, 0, arrivaltimecopy, 0, n);
System.arraycopy(burstTime, 0, burstTimecopy, 0, n);

for (int i = 0; i < n; i++) {
  totalTime += burstTime[i];
  if (arrivaltime[i] < min) {
    max = arrivaltime[i];
  }
}

for (int i = 0; i < n; i++) {
  if (arrivaltime[i] < min) {
    min = arrivaltime[i];
    minIndex = i;
    currentIndex = i;
  }
}
totalTime = min + totalTime;
tLap = min;
int tot = 0;
while (tLap < totalTime) {
  for (int i = 0; i < n; i++) {
    if (arrivaltimecopy[i] <= tLap) {
      if (priority[i] < priority[minIndex]) {
        minIndex = i;
        currentIndex = i;
      }
    }
  }
tLap = tLap + burstTimecopy[currentIndex];
```

```
      completionTime[currentIndex] = tLap;
      priority[currentIndex] = 9999;
      for (int i = 0; i < n; i++) {
        tot = tot + priority[i];
      }
    }
    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("\n*** Priority 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");

    }
    avgWT /= n;
    avgTAT /= n;
    System.out.println("\nAverage Wating Time: " + avgWT);
    System.out.println("Average Turn Around Time: " + avgTAT);
    sc.close();

  }
}
```

Output:

```
*** Priority Scheduling (Non Preemptive) ***
Enter Number of Process: 5
```

Enter Arrival Time for processor 1:0

Enter Burst Time for processor 1 : 3

Enter Priority for 1 process: 6

Enter Arrival Time for processor 2:15

Enter Burst Time for processor 2 : 20

Enter Priority for 2 process: 3

Enter Arrival Time for processor 3:2

Enter Burst Time for processor 3 : 1

Enter Priority for 3 process: 4

Enter Arrival Time for processor 4:5

Enter Burst Time for processor 4 : 9

Enter Priority for 4 process: 5

Enter Arrival Time for processor 5:10

Enter Burst Time for processor 5 : 8

Enter Priority for 5 process: 7

*** Priority Scheduling (Non Preemptive) ***

Processor	Arrival time	Brust time	Completion Time	Turn around time	Waiting time
P1	0ms	3ms	3ms	3ms	0ms
P3	2ms	1ms	5ms	3ms	2ms
P4	5ms	9ms	14ms	9ms	0ms
P5	10ms	8ms	22ms	12ms	4ms
P2	15ms	20ms	42ms	27ms	7ms

Average Wating Time: 2.6

Average Turn Around Time: 10.8