

### Priority Scheduling (Preemptive):

```
import java.util.Scanner;

class PriorityScheduling {

    public static void main(String[] args) {

        System.out.println("*** Priority Scheduling (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 + 1];
        int TAT[] = new int[n];
        int waitingTime[] = new int[n];
        int burstTimecopy[] = new int[n];
        int min = 0, count = 0;
        int temp, time = 0, end;
        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;
        }
        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(burstTime, 0, burstTimecopy, 0, n);
priority[n] = 999;
for (time = 0; count != n; time++) {
    min = n;
    for (int i = 0; i < n; i++) {
        if (arrivaltime[i] <= time && priority[i] < priority[min] && burstTime[i] > 0)
            min = i;
    }
    burstTime[min]--;
    if (burstTime[min] == 0) {
        count++;
        end = time + 1;
        completionTime[min] = end;
        waitingTime[min] = end - arrivaltime[min] - burstTimecopy[min];
        TAT[min] = end - arrivaltime[min];
    }
}

for (int i = 0; i < n; i++) {
    avgTAT += TAT[i];
    avgWT += waitingTime[i];
}

System.out.println("\n*** Priority Scheduling (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" + burstTimecopy[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 (Preemptive) \*\*\*

Enter Number of Process: 5

Enter Arrival Time for processor 1:0

Enter Burst Time for processor 1 : 4

Enter Priority for 1 process: 2

Enter Arrival Time for processor 2:1

Enter Burst Time for processor 2 : 3

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 4:3

Enter Burst Time for processor 4 : 5

Enter Priority for 4 process: 5

Enter Arrival Time for processor 5:4

Enter Burst Time for processor 5 : 2

Enter Priority for 5 process: 5

\*\*\* Priority Scheduling (Preemptive) \*\*\*

Processor	Arrival time	Brust time	Completion Time	Turn around time	Waiting time
-----------	--------------	------------	-----------------	------------------	--------------

P1	0ms	4ms	4ms	4ms	0ms
P2	1ms	3ms	7ms	6ms	3ms
P3	2ms	1ms	8ms	6ms	5ms
P4	3ms	5ms	13ms	10ms	5ms
P5	4ms	2ms	15ms	11ms	9ms

Average Wating Time: 4.4

Average Turn Around Time: 7.4