

All.java

```
import java.util.Scanner;

public class All {
    public static void main(String[] args){
        FCFS fcfs = new FCFS();
        SJF sjf = new SJF();
        Scanner scan = new Scanner(System.in);
        while(true){
            System.out.println("\n The available algorithms are: ");
            System.out.println("1. FCFS");
            System.out.println("2. SJF");
            System.out.println("3. Exit");
            System.out.println("Choose your algorithm: ");
            int algo = scan.nextInt();
            if (algo==1){
                System.out.println("Enter the number of processes: ");
                int n = scan.nextInt();
                int[] processes = new int[n];
                //Burst time of all processes
                int[] burst_time = new int[n];
                System.out.println("Enter the processes: ");
                for (int i = 0; i < n; i++){
                    processes[i] = scan.nextInt();
                }
                System.out.println("Enter the Burst time for he
processes: ");
                for (int i = 0; i < n; i++){
                    burst_time[i] = scan.nextInt();
                }
                fcfs.findavgTime(processes, n, burst_time);
            }
            else if(algo==2){
                System.out.println("Enter the number of processes: ");
                int n = scan.nextInt();
                int[] processes = new int[n];
                int[] burst_time = new int[n];
                int[] arr_time = new int[n];
                System.out.println("Enter the processes: ");
                for(int i = 0; i < n; i++){
                    processes[i] = scan.nextInt();
                }
                System.out.println("Enter the Burst Time of the
processes: ");
                for(int i = 0; i < n; i++){
                    burst_time[i] = scan.nextInt();
                }
                System.out.println("Enter the Arrival Time of the
processes: ");
                for(int i = 0; i < n; i++){
                    arr_time[i] = scan.nextInt();
                }
                Process[] proc = new Process[n];
            }
        }
    }
}
```

```

        for(int i = 0; i < n; i++){
            Process ind_pr = new Process(processes[i],
burst_time[i],
                arr_time[i]);
            proc[i] = ind_pr;
        }

        sjf.findavgTime(proc, proc.length);
    }
    else if(algo==3){
        System.out.println("Exiting the code...");
        break;
    }
    else{
        System.out.println("Invalid Input");
    }
}
scan.close();
}
}

```