Service Simulation

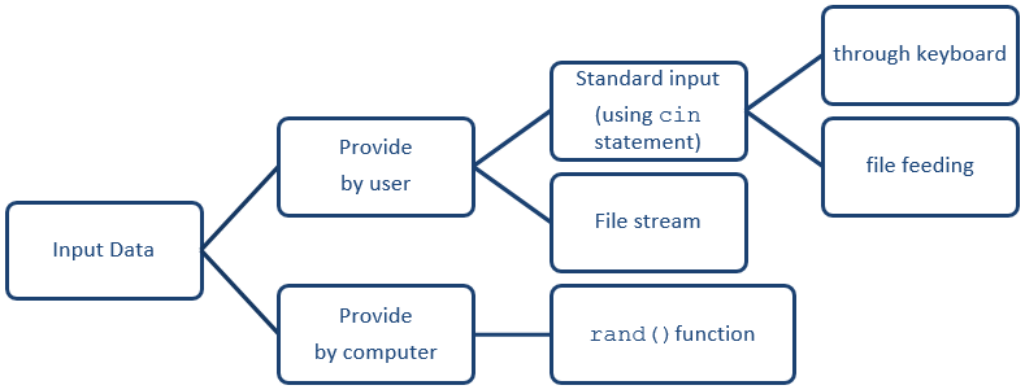
Important Points: rand(), Clock()

•To get the random value, we can either: use standard input (keyboard or file feeding) or file stream.randomly generated value using the rand() function.

Service simulation-complexity

- Service Simulation - many stages of complexity {server, service, customer}

Complexity	Simulation Characteristic	Example
1	One server with one service. One queue with many customers.	Queue at Car wash machine.
2	One server with many services. One queue with many customers.	Automated teller machine (ATM)
3	Many servers with one services. One queue with many customers.	*Can you think of one example?*
4	Many servers with many services. One queue with many customers.	*Can you think of one example?*
5	Many servers with many services. Many queue with many customers.	*Can you think of one example?*



- Assist by computer to get many information - customers (arrival time) and type of service and using
  - value randomly generated by rand() method

rand() method

- *rand* is a class defined in java.util.Random;.
- It is used to generate random numbers: integers, float, double, long, boolean
- generate random numbers, use methods - nextInt(), nextDouble(), nextLong() etc using that instance.

```
import java.util.*;

public class simulation {
          public static void main(String[] args) {
              Random rand = new Random();
              int n = rand.nextInt(10); // Obtain a number between [0 - 9].

               System.out.print(" " + n);
     }
}
```

Case: ABC Wash Machine

```
27          Clock i;
28          int nextArrival=0;
29          Random rand = new Random();
30
31          startTime.setTime(8,0,0);
32          endTime.setTime(8,30,0); // can change to 12 pm
33
34          for (i=startTime.getCopy();i.lessThan(endTime); ){
35                  nextArrival = rand.nextInt(10); nextArrival in the range 0 to 9
36                  i.addTimeMinute(nextArrival);
37                  if(i.lessThan(endTime)) {
38                     arrivalQueue.enqueue(i.getCopy());
39                     System.out.println("car arrival: " + i.toString() + " < " +
40                                                 endTime.toString());
41                  }
42          }
43
```

Clock()

- The class clock is a user defined class.
- The class Clock hold variables and methods to support and process instructions of time data type.
- class *Clock* comprises of:
  - 3 data members of type integer: hr, min and sec,
  - 11 member functions with public access: *setTime, getTime, printTime, incrementSeconds, incrementMinutes, incrementHours, equalTime, addTimeMinute, lessThan, earlier, durationSec*.

- Why Clock related to simulation?
  - Simulation process use time data type
  - Report statistical information – longest waiting **time**, record the arrival **time of customer etc**

```
import java.util.*;
public class MyQueue {
    public static void main(String args[]){
          Queue <Clock> q2 = new Queue<>();
          Clock myClock = new Clock();
           int m=25; int i=0;

           System.out.print("\nmyClock: " + myClock.toString());
           myClock.incrementHours();
           myClock.addTimeMinute(15);
           System.out.print("\nmyClock: " + myClock.toString());

           while (i<4){
              myClock.addTimeMinute(20);
              q2.enqueue(myClock.getCopy());
              i++;
           }

           System.out.print("\nq1: " + q2.toString());
    }
}
```

Case: ABC Wash Machine

```
18          //observation data
19          Clock totalWaitTime= new Clock();
20          Clock maxWaitTime = new Clock();
21          Clock totalServiceTime = new Clock();
22          Clock startTime= new Clock();
23          Clock endTime= new Clock();
24          Clock washEnd = new Clock();
25          Clock washStart = new Clock();
26
27          Clock i;
28          int nextArrival=0;
29          Random rand = new Random();
30
31          startTime.setTime(8,0,0);
32          endTime.setTime(8,30,0); // can change to 12 pm
33
34          for (i=startTime.getCopy();i.lessThan(endTime); ){
35                 nextArrival = rand.nextInt(10); nextArrival in the range 0 to 9
36                 i.addTimeMinute(nextArrival);
37                 if(i.lessThan(endTime)) {
38                    arrivalQueue.enqueue(i.getCopy());
39                    System.out.println("car arrival: " + i.toString() + " < " +
40                                                 endTime.toString());
41                 }
42          }
43
```