**JAVA LAB PROGRAM**

**Submitted by,**

Tamilvanan B.

2018503566.

MO Batch.

09-05-2020.

1. A Bank issues tokens to its customers that have 4 one-digit numbers 0 to 9 on it.The customers are called to the counter in any one of the following two choices:

a) A customer is attended if they have a token that has the digits in any order.

b) A customer is attended if they have a token that has the digits in required order.

Write a java program that displays the chances of customers being attended. Also, the program should track the number of customers attended. Few customers may have received the token but there is a possibility they leave the bank without going to the counter. Such customer’s token details should be removed.

Program the above scenario by incorporating the following concepts in your own way

Ø Classes & objects

Ø Constructors

Ø Garbage invocation request

Ø Static keyword

Ø Object array

**Program:**

import java.util.\*;

import java.util.concurrent.ThreadLocalRandom;

class customers{

String name;

int token,wait=0;

static int len=0;

static int overflow=100;

int[] count;

float probability\_crct,probability\_any;

customers(){

Scanner sc= new Scanner(System.in);

System.out.print("\nEnter the Customer Name: ");

this.name=sc.next();

System.out.print("Enter the Token number: ");

this.token=sc.nextInt();

this.count=new int[10];

for(int i=0;i<10;i++) this.count[i]=0;

int temp=this.token;

while(temp>0)

{

this.count[temp%10]++;

temp/=10;

}

this.probability\_crct=(float)1/10000;

this.wait=0;

len++;

}

customers(int temp){

this.token=temp;

this.count=new int[10];

for(int i=0;i<10;i++) this.count[i]=0;

while(temp>0)

{

this.count[temp%10]++;

temp/=10;

}

this.probability\_crct=(float)1/10000;

this.wait=0;

len++;

}

void get\_anyprob(){

this.probability\_any=24;

for(int i=0;i<10;i++)

{

if(count[i]==0) continue;

for(int j=1;j<=count[i];j++) this.probability\_any/=j;

}

this.probability\_any/=10000;

}

boolean match(int temp){

int[] t=new int[10];

for(int i=0;i<10;i++) t[i]=0;

while(temp>0)

{

t[temp%10]++;

temp/=10;

}

for(int i=0;i<10;i++){

if(t[i]!=this.count[i]) return false;

}

return true;

}

boolean due(){

this.wait++;

if(this.wait>=customers.overflow){

return true;

}

return false;

}

}

public class BankerProb{

public static void main(String args[]){

Scanner sc= new Scanner(System.in);

int n,i,j;

String name;

System.out.print("Enter the number of the customers in the queue: ");

n=sc.nextInt();

customers[] arr=new customers[n];

for(i=0;i<n;i++)

arr[i]=new customers();

for(i=0;i<n;i++)

{

arr[i].get\_anyprob();

System.out.println("\nCustomer Name: " + arr[i].name + "\nCustomer Token: " + arr[i].token);

System.out.println("Required Order Probability: " + arr[i].probability\_crct + " and Any order Probability: " + arr[i].probability\_any);

}

System.out.print("\n");

int trial=0;

while(customers.len>0){

j=ThreadLocalRandom.current().nextInt();

if(j<0) j\*=-1;

j%=10000;

trial++;

for(i=0;i<n;i++){

if(arr[i]!=null && arr[i].match(j)){

arr[i]=null;

System.gc();

customers.len--;

break;

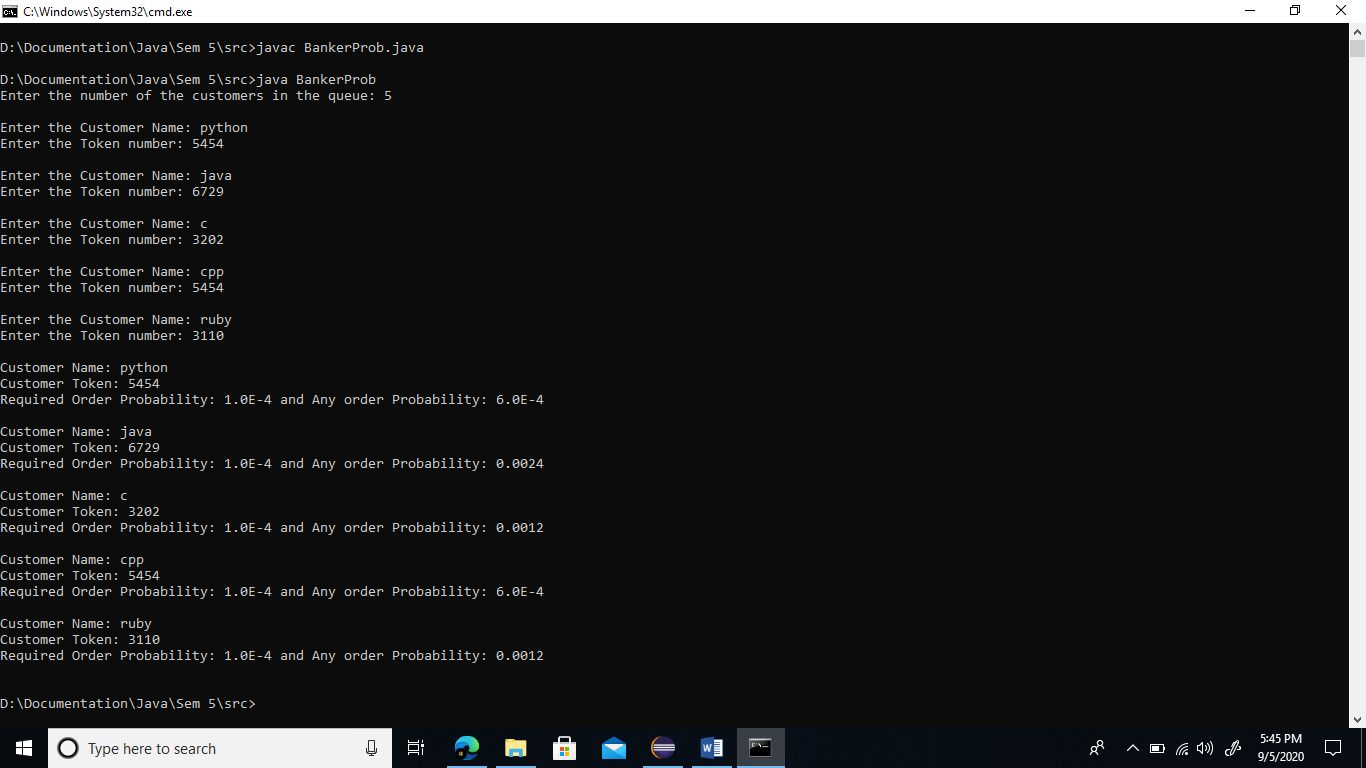
}

}

}

}

}

**Output:**