ASSIGNMENT NO: 06

Coffee Shop Line (Simple Queue) OR Printer Spooler (Circular Queue)

Name: Munajja Mujafar Dalimbkar
PRN: B25CE2011

PROBLEM STATEMENT:
Coffee Shop Line (Simple Queue):
Arrival: Customers arrive at the coffee shop and stand in line. Order Processing: The first customer in line gets their order taken, and the barista starts making the coffee.Serving: Once the first customer is served, they leave the queue, and the next customer in line moves forward to be served. Write a program to implement a simple queue.

CODE:

```
#include<iostream>

using namespace std;

class CoffeeShop

{

static const int MAX = 10;

   int tokens[MAX];

int nextToken;


int rear;

int front ;


public:

CoffeeShop()

{
```

```
front = -1;
rear = -1;
nextToken = 0;
}

int isEmpty()
{
if(front ==-1 || front == MAX || rear < front)
{ return 1; }

else
{ return 0; }
}
   int isFull()
   {
if(rear == MAX-1)
{ return 1; }
else
{ return 0; }
}

   void enqueue() {
```

```
        if (isFull())
        {
            cout<< "Queue is full."<<endl;
        }
        else
        {
          if(rear==-1) {front=0;}
            rear = rear + 1;
            tokens[rear] = nextToken;
            cout << "Customer given token number: " << nextToken << "\n";
            nextToken++;
        }
    }

   void dequeue()
    {
if (isEmpty())
{
cout<<"Queue is Empty."<<endl;
}
```

```
else

{

cout<<"Token processed number:"<<tokens[front]<<endl;

front = front + 1;

}

}


};


int main()

{

CoffeeShop c;

int choice;

char ch = 'Y';

do{

cout<<"Enter your choice from below:";

cout<<"\n1.Get Token \n2.Process Token \n3.Exit"<<endl;

cin>> choice;

switch(choice)

{

case 1:
```

```
c.enqueue();

break;

case 2:

c.dequeue();

break;

case 3:

cout<<"Exiting program..."<<endl;

break;

}

cout<<"Do you want to continue?(Y/N)";
cin>>ch;
}while(ch=='Y');

return 0;

}
```

OUTPUT:

```
Enter your choice from below:
1.Get Token
2.Process Token
3.Exit
1
Customer given token number: 0
Do you want to continue?(Y/N)N


=== Code Execution Successful ===
```