**TITLE:-** Implementation and Usage of queue

**Objective:-** Development of ticket booking module for a day of typical Cinema hall..

**THEORY:-**

• DSA : A data structure Is a named location that is used to store and organize data. And, an algorithm is a collection of steps to solve a particular problem. Learning data structures and algorithms allow us to write optimized and efficient computer programs.

•QUEUE : In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence.

**Implementation:**

#include <iostream>

#include <fstream>

using namespace std;

//creating Node for linked list

template<typename T>

class Node{

public:

T ticketType;

T phoneNumber;

T arrivalTime;

Node<T>\* next;

Node(T ticketType,T phoneNumber,T arrivalTime){

this->ticketType=ticketType;

this->phoneNumber=phoneNumber;

this->arrivalTime=arrivalTime;

this->next=NULL;

}

};

//creating Queue

template<typename T>

class Queue{

int size;

Node<T>\*head;

Node<T>\*tail;

public:

Queue(){

head=NULL;

tail=NULL;

size=0;

}

int getSize(){

return size;

}

void enqueue(T ticketType,T phoneNumber,T arrivalTime){

Node<T>\*node=new Node(ticketType,phoneNumber,arrivalTime);

if(head==NULL){

head=node;

tail=node;

} else{

tail->next=node;//inserting at tail

tail=tail->next;

}

size++;

}

bool isEmpty(){

return size==0;

}

T front(){

return head->data;

}

void dequeue(){

if(!isEmpty()){

Node<T>\*tempHead=head;

head=head->next;

tempHead=NULL;

delete tempHead;

}else{

cout<<"Queue is Empty";

}

}

};

//class for store customer data;

class Customer{

private:

string ticketType;

long phoneNumber;

long arrivalTime;

public:

//constructor for adding data in customer

Customer(string ticketType,long phoneNumber,long arrivalTime){

this->ticketType=ticketType;

this->phoneNumber=phoneNumber;

this->arrivalTime=arrivalTime;

}

void showTicket(){

cout<<ticketType;

}

void showPhoneNumber(){

cout<<phoneNumber;

}

void showArrivalTime(){

cout<<arrivalTime;

}

void setTicketType(string ticketType){

this->ticketType=ticketType;

}

void setPhoneNumber(long phoneNumber){

this->phoneNumber=phoneNumber;

}

void setArrivalTime(long arrivalTime){

this->arrivalTime=arrivalTime;

}

};

int main(){

fstream file;

file.open("../Lab/ticketData.txt",ios::in);

if(file.is\_open()){

string data;

long phoneNumber,arrivalTime;

string ticketType;

getline(file,data,',');

getline(file,ticketType,',');

cout<<data;

cout<<endl<<ticketType;

} else{

cout<<"file is not open";

}

file.close();

return 0;

}

**ticketData.txt**

Phone Number,Ticket Type,Arrival Time

+977-9849023236,Individual,10 Unit

+977-9849023260,Family,21 Unit

+977-9849023299,Family,22 Unit

+977-9849023220,Individual,24 Unit

+977-9849023243,Family,27 Unit

**DISCUSSSIN AND CONCLUSION:**

**Here,** we got to know the implementation of given real problem by using Queue

Hence, we became familiar with Queue by using real world problem