|  |  |  |
| --- | --- | --- |
| https://upload.wikimedia.org/wikipedia/commons/thumb/4/4e/VU_Logo.png/260px-VU_Logo.png | Data Structures (CS301)  Assignment # 02  Semester Spring 2021 | Total Marks = 20  Deadline Date  02/06/2021 |
| Please carefully read the following instructions before attempting assignment.  RULES FOR MARKING  It should be clear that your assignment would not get any credit if:   * The assignment is submitted after the due date. * The submitted assignment does not open or file is corrupt. * Strict action will be taken if submitted solution is copied from any other student or from the internet.   Lectures:   * Lectures 9 to 15 are covered in this assignment | | |
|  | | |

**Problem Statement:**

Due to the Coronavirus pandemic Government has asked all the major grocery stores to serve the customers as soon as possible so that the gathering of the people may be avoided. In this regard all the grocery stores have decided that there will be two queues for the customers at each grocery store. One queue is the express queue and second is the normal queue. Express queue will have the customers which have bought 5 or less than 5 grocery items. Similarly normal queue will have the customers with grocery items greater than 5. These queues will help the customer to choose the line according to their grocery shopping requirements and they will have to wait in their queue accordingly. The idea is to develop a C++ program that could serve the purpose of FIFO (first come first served)for both express and normal service queues so that each customer may select queue according to his/her shopping requirements. The program must have the features to:

* Record the information of the customer (name, shopping information etc.) and identify the type of customer so that he/she may select express queue or normal queue)
* Simulate both express and normal queues i.e., count the number of customers and the serving time of each customer in each queue.

The program must have the following functional features:

1. Record information of each customer that would be the customer name, customer id, number of grocery items purchased.
2. Send each customer to either express queue or normal queue based on the number of grocery items customer has purchased.
3. Keep the count of customers in each queue. Whenever a customer enters or exits a queue display the total number of customers in that queue
4. Print the information of the customers of both the queues.

**Furthermore:** You need to consider following necessary steps while developing the solution.

* Use linked list to make queue of customers. Here CustomerNode class will be used in place of Node class.
* Create a single CustomerQueue class to handle both express and normal queues customers. Use addCustomer(enqueue), removeCustomer(dequeue) and methods for smooth operations of queue.
* Collect all information of customers like
  + Customer Name, Customer ID, Number of grocery items purchased,
* After getting required information of customer, check the number of grocery items customer has purchased and then put him in the respective queue.
* Whenever a customer enters into a queue print the message “Customer entered in express/normal queue” and also print the total number of customers in that queue. Use removeCustomer() method to remove customer one by one and display the information of that customer along with the remaining customers count of each queue.

**Solution:**

#include <iostream>

#include <conio.h>

// defining class

using namespace std;

class CustomerNode

{

public:

int customer\_id;

int item\_purchased;

string customer\_name;

CustomerNode \*left;

CustomerNode \*right;

CustomerNode(int id, int item, string name)

{

customer\_id = id;

item\_purchased = item;

customer\_name = name;

}

};

class queue

{

public:

CustomerNode \*arr[100];

int n = 100;

int front = -1;

int rear = -1;

int count = 0;

void addCustomer(CustomerNode \*customer)

{

if (front == -1)

{

front = 0;

rear++;

}

arr[rear] = customer;

count++;

}

void recoverCustomer()

{

if (count == 0)

{

cout << "Queue is empty" << endl;

cout << "" << endl;

}

else

{

cout << "Customer Name: " << arr[front]->customer\_name << endl;

cout << "Customer ID: " << arr[front]->customer\_id << endl;

// Your truly helper

cout << "Number of Items Purchased: " << arr[front]->item\_purchased << endl;

cout << "" << endl;

front++;

count--;

}

}

};

int main()

{

queue normalQueue;

queue expressQueue;

int choice = 0;

int temp;

do

// functions perform

{

cout << "---CUSTOMER INFORMATION---" << endl;

cout << " ---Press 1 to Add Customer---" << endl;

cout << " ---Press 2 to Remove Customer---" << endl;

cout << " ---Press 3 to exit---" << endl;

cout << "" << endl;

cin >> choice;

if (choice == 1)

{

// saving record of id, name and item

int id;

int item;

string name;

cout << "Enter Name: ";

cin >> name;

cout << "Enter Customer ID: ";

cin >> id;

// free to change assignment on vuanswer.com

cout << "Enter Number of Items: ";

cin >> item;

cout << "" << endl;

// ( FREE ASSIGNMENT PROVDE BY VU ANSWER)

CustomerNode \*newcustomer = new CustomerNode(id, item, name);

// detail save in express queue

if (item <= 5)

{

expressQueue.addCustomer(newcustomer);

{

cout << "Customer Entered into Express Queue (VUAnswer.com)" << endl;

cout << "---Customer in Express Queue are: " << expressQueue.count << endl;

cout << "" << endl;

}

}

// moreover detail save in normal queue

else

{

normalQueue.addCustomer(newcustomer);

cout << "Customer is Entered into Normal queue" << endl;

// dear student change your assignment freely just whatsapp +923162965677

cout << "Customer in Normal Queue are: " << normalQueue.count << endl;

cout << "" << endl;

}

}

else if (choice == 2)

{

int choice;

cout << "To Remove Customer from Express Queue press 1 " << endl;

cout << "To Remove Customer from Normal Queue press 2 " << endl;

cin >> choice;

// solution provide by vu answer

cout << "" << endl;

if (choice == 1)

{

int choice;

cout

<< "Information of Remove Customer is: " << endl;

expressQueue.recoverCustomer();

// free to come for correct assignment on vuanswer.com

cout << "Remaining Customer Queue are: " << expressQueue.count << endl;

}

if (choice == 2)

{

int choice;

cout << "Information of Remove Customer is: " << endl;

normalQueue.recoverCustomer();

cout << "Remaining Customer Queue are: " << normalQueue.count << endl;

}

}

} while (choice != 0);

}

**Output Screen Shot:**

