CSC248 – Fundamentals of Data Structure
Academic Session October 2023 – February 2024
Lab Assignment 6 - Queue

Course Outcomes (CO)	LO1	LO2	LO3
CO1			
CO2	√	√	√
CO3			

1.1 Given the following `Queue` ADTs:

```
public class House
{
      public String type; //ex: Semi-D, Terrace
      public String location;
      public double size;
      public double price;

      public House(String t, String l, double s, double p)
      {…}
      public String getType() {…}
      public String getLocation() {…}
      public double getSize() {…}
      public double getPrice() {…}
}

public class Queue
{
      public Queue()  {…}
      public void enqueu(Object elem) {…}
      public Object dequeue()  {…}
      public boolean isEmpty() {…}

   //definition for other methods
}
```

Write a Java application to solve the following problems **(use built-in method-LinkedList)**.

a) Create a Queue object named as `qHouse`.
b) Input ten (10) objects of houses and store them into `qHouse`.
c) Get all houses from `qHouse` and store all type of semi-D houses into a queue called q`Semi_D` and all terrace houses into a queue called `qTerrace`. At the end of process, all houses must remain in an original queue, `qHouse`.
d) Display the information of house from `qTerrace` that the price is less than RM150,000. At the end of process, all houses must remain in an original queue, `qTerrace`.
e) Count the number of houses that the price is more than RM 300,000.00 and display all information for that houses from `qHouse`.

1.2 Given a declaration of queue structure as follows:

```
public class Customer
{
  private String name;
  private int accountNo;
  private double saving;
  private double totalTransaction;

  // declaration for another methods
  public Customer(String, int, double, double){…}
  public String getName(){…}
  public int getAccountNo(){…}
  public double getSaving(){…}
  public double getTotalTransaction(){…}
  public String toString(){…} //print customer information
  public boolean process(){…}//return TRUE if qualified(saving
                          //must exceed RM 1000 after transaction)
                           //or FALSE if disqualified
}

public class Node
{
  Object data;
  queueNode link;

  // create a queueNode that refers to object elem
  ListNode(Object elem);
  // create a queueNode that refers to object elem and to the
  // next queueNode in the list
  ListNode(Object elem, queueNode nextElem);
  //return a reference to the object in this node
  Object getData();
  //return the next node
  queueNode getLink();
}

public class ListNode{
  private  queueNode first;
  private queueNode last;

  //definition for other methods

}

public class QUEUE extends ListNode
{
  public QUEUE();
  // insert new element into queue
  public void enqueue(Object elem);
  // to delete queue element
  public Object dequeue();
  public bool isEmpty ();           //to determine empty list
  public Object getFirst();         //return first element
  public Object getNext();          //return the next element
  public Object getLast();          //return the last element
}
```

i) Based on the declaration of class ***Customer, Node, ListNode*** and ***QUEUE***, answer the following questions:

a) Write a complete program for all methods and classes **(use UDT)**.

b) Write an application program to do the following tasks:

i. Declare a **QUEUE** object named as ***qCustomer*** to store customer information data.

ii. Declare a **QUEUE** object named as ***qQualify***, to store customer information data that qualified to apply a loan

iii. Assume a queue for ***qCustomer*** has been inserted with some values (please insert some data by the user). Determine either any customers is qualify or disqualify to apply for a loan. If there are any customers qualified to apply for a loan, store the information into ***qQualify***.

iv. Print all customer information from ***qQualify*** list.