NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

Practicum 1 :

```
package Practicum1;

public class Queue08 {
    int[] data;
    int front, rear, size, max;

    public Queue08(int n){
        max = n;
        data = new int[max];
        size = 0;
        front = rear = -1;
    }

    public boolean IsEmpty(){
        if (size == 0){
            return true;
        } else {
            return false;
        }
    }

    public boolean IsFull(){
        if (size == max){
            return true;
        } else {
            return false;
        }
    }

    public void clear(){
        if (!IsEmpty()){
            front = rear = -1;
            size = 0;
            System.out.println("Queue was sucsessfully emptied");
        } else {
            System.out.println("Queue still empty");
        }
    }

    public void Enqueue(int dt){
        if (IsFull()){
            System.out.println("Queue is Full");
        } else {
            if (IsEmpty()){
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
                front = rear = 0;
            } else {
                if (rear == max - 1){
                    rear = 0;
                } else {
                    rear++;
                }
            }
            data[rear] = dt;
            size++;
        }
    }

    public int Dequeue(){
        int dt = 0;
        if (IsEmpty()){
            System.out.println("Queue is empty");
        } else {
            dt = data[front];
            size--;
            if (IsEmpty()){
                front = rear = -1;
            } else {
                if (front == max - 1){
                    front = 0;
                }else{
                    front++;
                }
            }
        }
        return dt;
    }

    public void peek(){
        if (!IsEmpty()){
            System.out.println("Front Element: " + data[front].account + " " +
data[front].name + " " + data[front].address + " " + data[front].age + " " +
data[front].balance);
        } else {
            System.out.println("Queue is empty");
        }
    }

    public void print(){
        if (IsEmpty()){
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
            System.out.println("Queue is enply");
        } else {
            int i = front;
            while (i != rear) {
                System.out.println(data[front].account + " " + data[front].name + "
" + data[front].address + " " + data[front].age + " " + data[front].balance);
                i = (i + 1)%max;
            }
            System.out.println(data[i] + " ");
            System.out.println("number of elemets (size) = " +size);
        }
    }
}
```

```
package Practicum1;
import java.util.Scanner;
public class QueueMain08 {
    public static void menu() {
        System.out.println("Input Queue Operation:");
        System.out.println("1.Enqueue");
        System.out.println("2.Dequeue");
        System.out.println("3.Print");
        System.out.println("4.Peek");
        System.out.println("5.Clear");
        System.out.println("==================");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Input queue capacity(Max):");
        int n = sc.nextInt();

        Queue08 Q = new Queue08(n);
        int choose;

        do{
            menu();
            choose = sc.nextInt();
            switch (choose){
                case 1:
                    System.out.println("Input New Data: ");
                    int InputData = sc.nextInt();
                    Q.Enqueue(InputData);
                    break;
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
                case 2:
                    int DeletedData = Q.Dequeue();
                    if (DeletedData != 0) {
                        System.out.println("Deleted data: " +DeletedData);
                        break;
                    }
                case 3:
                    Q.print();
                    break;
                case 4:
                    Q.peek();
                    break;
                case 5:
                    Q.clear();
                    break;
            }
        } while (choose == 1 || choose == 2 || choose == 3 || choose == 4 || choose
== 5);
    }
}
```

Question :

1. In the constructor, why is the initial value of the front and rear attributes is -1, while the size attribute is 0?
   Answer : because it doesn't show any data
2. On Enqueue method , explain the purpose of the following code!
   Answer : If the queue is still empty, it means that the data that will enter will be the first data and also the last data in the queue, which is at index position 0
3. On Dequeue method , explain the purpose of the following code!
   Answer : If front is at the last index of the array then put index 0
4. On print method, why in loop process variable i does not start from 0 (int i=0), but int i=front?
   Answer : Because looping doesn't always start at index 0
5. Pay attention to the print method, explain the meaning of the following code!
   Answer :
6. Show the code that is the queue overflows !

```
BankCustomer08(String account, String name, String address, int age, double
balance){
        this.account = account;
        this.name = name;
        this.address = address;
        this.age = age;
        this.balance = balance;
    }
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

7. When there is a queue overflow and queue underflow conditions , the program can still run and only display text information. Modify the program : when a queue overflow and queue underflow occurs, the program will terminated!
   Answer: If an addition is made to the queue but the elements in the queue reach the maximum capacity of the queue, it is called overflow. If an element is taken but there is no element in the queue, it is called underflow

Practicum 2 :

```
package Practicum2;

public class BankCustomer08 {
    BankCustomer08(String account, String name, String address, int age, double
balance){
        this.account = account;
        this.name = name;
        this.address = address;
        this.age = age;
        this.balance = balance;
    }
}
```

```
package Practicum2;

public class Queue08 {
    BankCustomer08[] data;
    int front, rear, size, max;

    public Queue08(int n){
        max = n;
        data = new BankCustomer08[max];
        size = 0;
        front = rear = -1;
    }

    public boolean IsEmpty(){
        if (size == 0){
            return true;
        } else {
            return false;
        }
    }

    public boolean IsFull(){
        if (size == max){
            return true;
        } else {
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
            return false;
        }
    }

    public void clear(){
        if (!IsEmpty()){
            front = rear = -1;
            size = 0;
            System.out.println("Queue was sucsessfully emptied");
        } else {
            System.out.println("Queue still empty");
        }
    }

    public void Enqueue(BankCustomer08 dt){
        if (IsFull()){
            System.out.println("Queue is Full");
        } else {
            if (IsEmpty()){
                front = rear = 0;
            } else {
                if (rear == max - 1){
                    rear = 0;
                } else {
                    rear++;
                }
            }
            data[rear] = dt;
            size++;
        }
    }

    public BankCustomer08 Dequeue(){
        BankCustomer08 dt = new BankCustomer08();
        if (IsEmpty()){
            System.out.println("Queue is empty");
        } else {
            dt = data[front];
            size--;
            if (IsEmpty()){
                front = rear = -1;
            } else {
                if (front == max - 1){
                    front = 0;
                }else{
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
                    front++;
                }
            }
        }
        return dt;
    }

    public void peek(){
        if (!IsEmpty()){
            System.out.println("Front Element: " + data[front].account + " " +
data[front].name + " " + data[front].address + " " + data[front].age + " " +
data[front].balance);
        } else {
            System.out.println("Queue is empty");
        }
    }

    public void print(){
        if (IsEmpty()){
            System.out.println("Queue is enply");
        } else {
            int i = front;
            while (i != rear) {
                System.out.println(data[i].account + " " + data[i].name + " " +
data[i].address + " " + data[i].age + " " + data[i].balance);
                i = (i + 1)%max;
            }
            System.out.println(data[i].account + " " + data[i].name + " " +
data[i].address + " " + data[i].age + " " + data[i].balance);
            System.out.println("number of elemets (size) = " +size);
        }
    }
}
```

```
package Practicum2;

import java.util.Scanner;

public class QueueMain08 {
    public static void menu(){
        System.out.println("Select Menu:");
        System.out.println("1.Add new data to Queue");
        System.out.println("2.Remove data / Exit from Queue");
        System.out.println("3.The head / Front of the Queue");
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
        System.out.println("4.Display All Queue Data");
        System.out.println("========================");
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Input queue capacity(Max): ");
        int max = sc.nextInt();
        Queue08 qObject = new Queue08(max);

        int choose;
        do{
            menu();
            choose = sc.nextInt();
            sc.nextLine();
            switch(choose){
                case 1:
                    System.out.println("Bank Account: ");
                    String bAccount = sc.nextLine();
                    System.out.println("Name: ");
                    String name = sc.nextLine();
                    System.out.println("Address: ");
                    String address = sc.nextLine();
                    System.out.println("Age: ");
                    int age = sc.nextInt();
                    System.out.println("Balance: ");
                    double balance = sc.nextDouble();
                    BankCustomer08 mb = new BankCustomer08(bAccount, name, address,
age, balance);
                    sc.nextLine();
                    qObject.Enqueue(nb);
                    break;
                case 2:
                    BankCustomer08 data = qObject.Dequeue();
                    if (!"".equals(data.account) && !"".equals(data.name) &&
!"".equals(data.address) && data.age != 0 && data.balance != 0){
                        System.out.println("Element removed from the queue: " +
data.account + " " + data.name + " " + data.address + " " + data.age + " " +
data.balance);
                        break;
                    }
                case 3:
                    qObject.peek();
                    break;
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
                case 4:
                    qObject.print();
                    break;
            }
        } while (choose == 1 || choose == 2 || choose == 3 || choose == 4);
    }
}
```

Question :

1. In QueueMain00 class , explain the fuction of IF syntax in the following code!
   Answer: to provide conditions
2. Modify the program by adding a new method called peekRear to the class Queue00, this class is used to check the queue that is in the last position! Also add to menu list 5. Check the last queue in class QueueMain00 so method peekRear can be called!
   Answer:
   public static Object peekRear(){
   if(front==Queue.length){
   Queue=new Object[3];
   } return Queue[rear];
   }

Exercise:

1. Add the following two methods to the class Queue00 at Practicum
   a. Method peekPosition (data: int): void To display the position of a data in the queue , for example by sending certain data, the position (index) of the data will be known.

```
 public void peekPosition(){
       Scanner scan = new Scanner(System.in);
        System.out.println("Enter size of array");
        int N = scan.nextInt();
        int[] arr = new int[N + 2];
    System.out.println("All Peak Elements : ");
        for (int i = 1; i <= N; i++)
            if (arr[i - 1] <= arr[i] && arr[i] >= arr[i + 1])
                System.out.println(arr[i] +" at position "+ i);

        System.out.println();
    }
}
```

   b. Method peekAt (position : int) : void To display data that is at a certain position (index) Adjust the menu list in the class QueueMain00 so that both methods can be called!

```
public static void findPeek(){
        if (n == 1)
            return 0;
        if (arr[0] >= arr[1])
                return 0;
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
        if (arr[n - 1] >= arr[n - 2])
            return n - 1;
    for(int i = 1; i < n - 1; i++){
        if (arr[i] >= arr[i - 1] &&
            arr[i] >= arr[i + 1])
            return i;
    }return 0;
    }

    public static void main(String[] args) {
        int arr[] = { 1, 3, 20, 4, 1, 0 };
        int n = arr.length;
        System.out.print("Index of a peak point is " +
            findPeak(arr, n));
    }
```

2. Make a queue program to illustrate students who are asking for a KRS signature from a DPA lecturer on campus. When a student is going to queue , he must first write down his NIM, name, absence, and GPA as illustrated in the following class diagram:

```
3. package Exercise2;
4.
5. import java.util.Scanner;
6.
7. public class QueueStd {
8.     int num[MAX];
9.     int head=-1, tail=-1, k=1;
10.
11.    boolean IsEmpty(){
12.        if(tail == -1){
13.            return true;
14.        } else {
15.            return false;
16.        }
17.    }
18.
19.    boolean IsFull(){
20.        if(tail == MAX-1){
21.            return true;
22.        } else {
23.            return false;
24.        }
25.    }
26.
27.    void Enter(int ya){
28.        if(IsEmpty()){
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
            head=tail=0;
        } else {
            tail++;
        }
        num[tail]=ya;
    }

    void Out(){
        if(IsEmpty()){
            System.out.println("Queue is empty");
        } else {
            for(int j=head;j<tail;j++){
                num[j]=num[j+1];
            }
            tail--;
            if(tail==-1){
                head=-1;
            }
        }
    }

    void Clear(){
        head=tail=-1;
        k=1;
    }

    void View(){
        if(IsEmpty()){
            System.out.println("Queue is empty");
        } else {
            int j;
            System.out.println("Queue registered");
                for(int k=1;k<=j;k++){
                    System.out.println("Queue num:" +k);
                    System.out.println("Queue delayed");
                } for(j=head;j<=tail;j++){
                System.out.println("Queue num:" +num[j]);
                }
            }
        }
    }

    int main(){
        int choice,order;
        int cls;
```

NAME : FARICHA AULIA
NIM : 2141720155
ABSENT : 08
CLASS : 1I - IT
TOPIC : JOBSHEET 10 QUEUE

```
74.         do{
75.             menu();
76.             cls = sc.nextInt();
77.             switch (cls){
78.                 case 1:
79.                     System.out.println("Input New Data: ");
80.                     int InputData = sc.nextInt();
81.                     Q.Enqueue(InputData);
82.                     break;
83.                 case 2:
84.                     int DeletedData = Q.Dequeue();
85.                     if (DeletedData != 0) {
86.                         System.out.println("Deleted data: " +DeletedData);
87.                         break;
88.                     }
89.                 case 3:
90.                     Q.Out();
91.                     break;
92.                 case 4:
93.                     Q.enter();
94.                     break;
95.                 case 5:
96.                     Q.clear();
97.                     break;
98.             }
99.         }
100.              while (cls == 1 || cls == 2 || cls == 3 || cls == 4 || cls ==
    5);
101.             }
102.         }
103.     }
104.
```