INFORMATICS
INSTITUTE OF
TECHNOLOGY

ROBERT GORDON
UNIVERSITY ABERDEEN

Informatics Institute of Technology

B.Sc. (Hons) Artificial Intelligence and Data Science

Module: CM1602 Data Structures and Algorithms for Artificial Intelligence

Module Coordinator: Mr. Shivaraam Raghu
Ms. Niwarthana Kariyabaduge

Coursework Report

Student Details: 20210537 2117526 Dineth Hasaranga

Task 1

(a)

Queue data structure.

Reason:

The applications when placed in the order which they were submitted, can be accessed one at a time in the same order which they were submitted. Which means the customer which submitted the application first will be served first. New customers will be added to the rear of the line. Since queue is a FIFO (First In First Out) above requirement is fulfilled.

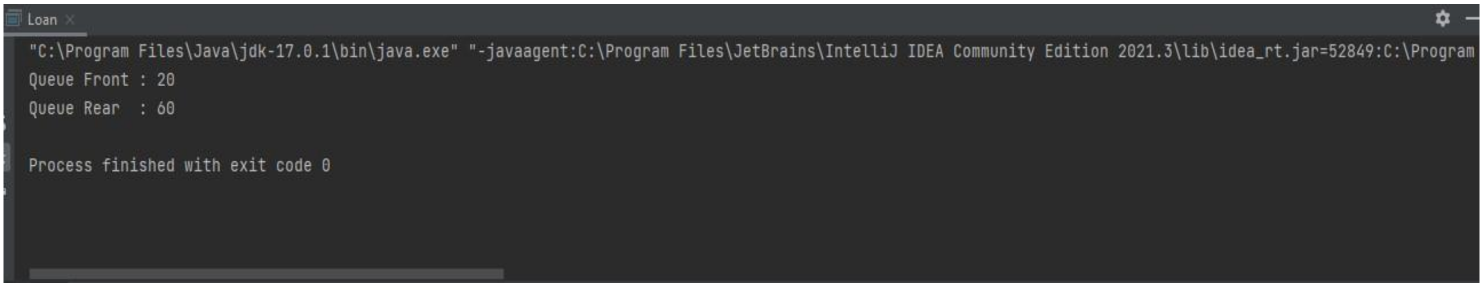
Also, I used the Queue with linked list implementation because the loan module must be able to handle any number of loan applications. Since linked lists are self-resizable at runtime the above requirement is fulfilled.

(b)**CLASS 1:**

```
package com.company;

class QueueNode {
    int key;
    QueueNode next;

    public QueueNode(int key) {
        this.key = key;
        this.next = null;
    }

}

class Queue {
    QueueNode front, rear;

    public Queue(){
        this.front = this.rear = null;
    }

    void enqueue(int key) {
        QueueNode map = new QueueNode(key);

        if (this.rear == null) {
            this.front = this.rear = map;
            return;
        }
        this.rear.next = map;
        this.rear = map;
    }

    void dequeue() {
        if (this.front == null)
            return;

        QueueNode map = this.front;
        this.front = this.front.next;

        if (this.front == null)
            this.rear = null;
    }
}
```

CLASS 2:

```
package com.company;

public class Loan {
    public static void main(String[] args) {
        Queue q = new Queue();

        q.enqueue(10);
        q.enqueue(20);
        q.dequeue();
        q.enqueue(40);
        q.enqueue(60);


        System.out.println("Queue Front : " + q.front.key);
        System.out.println("Queue Rear  : " + q.rear.key);

    }
}
```



```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3\lib\idea_rt.jar=52849:C:\Program
Queue Front : 20
Queue Rear  : 60

Process finished with exit code 0
```

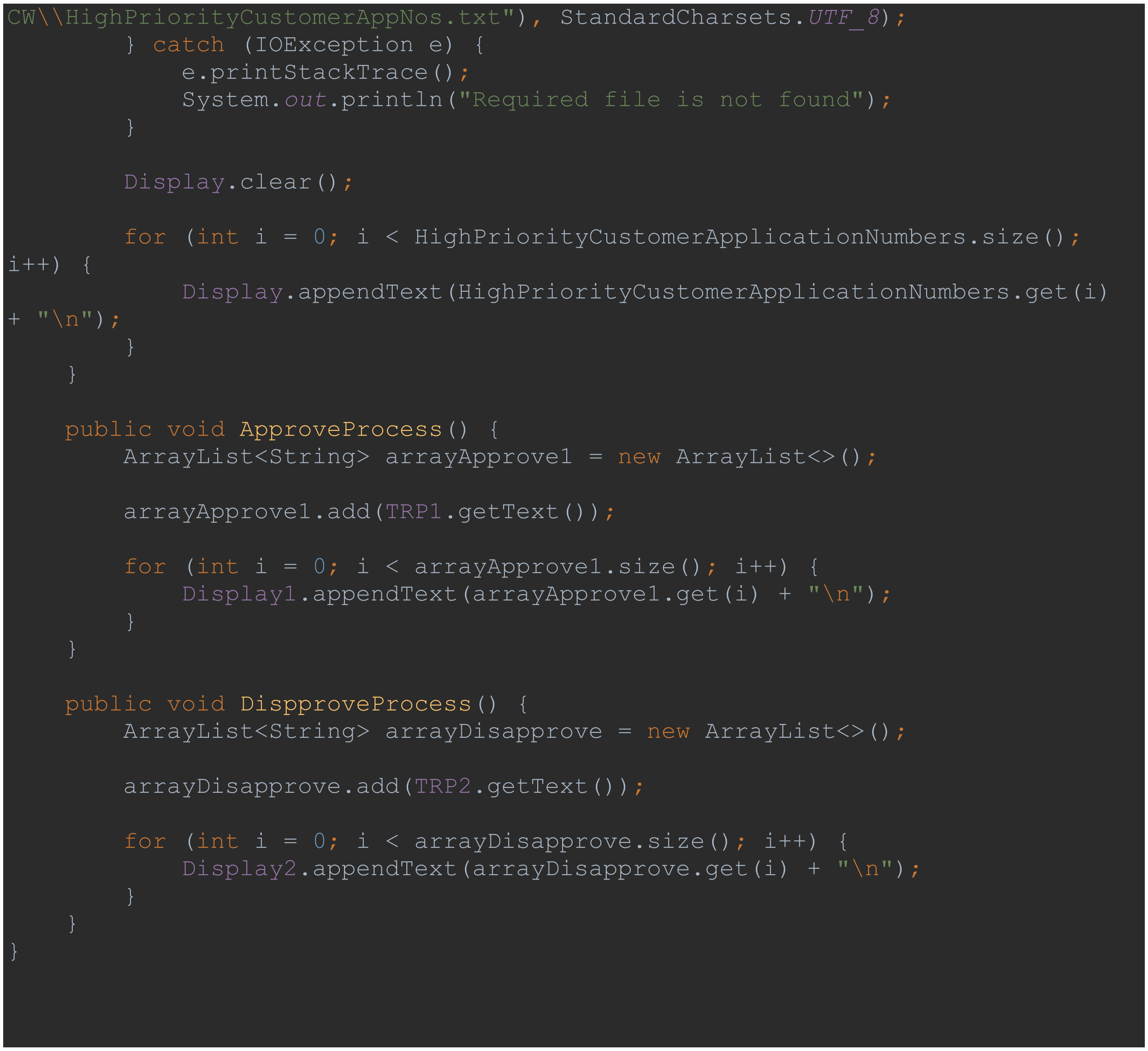
(c)Loan System Implementation: (GUI)

QueueNode Class:

```
package sample;

class QueueNode {
    int applicationNumber;
    String NIC;
    long accountNumber;
    String loanType;
    String reasonOfLoan;
    String collateralDescription;
    QueueNode next;

    public String getNIC() {
        return NIC;
    }

    public void setNIC(String NIC) {
        this.NIC = NIC;
    }

    public long getAccountNumber() {
        return accountNumber;
    }

    public void setAccountNumber(int accountNumber) {
        this.accountNumber = accountNumber;
    }

    public String getLoanType() {
        return loanType;
    }

    public void setLoanType(String loanType) {
        this.loanType = loanType;
    }

    public String getReasonOfLoan() {
        return reasonOfLoan;
    }

    public void setReasonOfLoan(String reasonOfLoan) {
        this.reasonOfLoan = reasonOfLoan;
    }

    public String getCollateralDescription() {
        return collateralDescription;
    }

    public void setCollateralDescription(String collateralDescription) {
        this.collateralDescription = collateralDescription;
    }


    public QueueNode(int applicationNumber, String NIC, int accountNumber,
String loanType, String reasonOfLoan, String collateralDescription) {
        this.applicationNumber = applicationNumber;
        this.NIC = NIC;
        this.accountNumber = accountNumber;
        this.loanType = loanType;
        this.reasonOfLoan = reasonOfLoan;
        this.collateralDescription = collateralDescription;
        next = null;
    }
}

class Queue {
    QueueNode front, rear;

    public Queue(){
        this.front = this.rear = null;
    }

    void enqueue(int applicationNumber,String NIC, int accountNumber, String
loanType, String reasonOfLoan,String collateralDescription) {
        QueueNode map = new
QueueNode(applicationNumber,NIC,accountNumber,loanType,reasonOfLoan,collatera
lDescription);

        if (this.rear == null) {
            this.front = this.rear = map;
            return;
        }
        this.rear.next = map;
        this.rear = map;
    }

    void dequeue() {
        if (this.front == null)
            return;

        QueueNode map = this.front;
        this.front = this.front.next;

        if (this.front == null)
            this.rear = null;
        return;
    }

}
```

HighPriorityCusEmployee Class:

```
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;

public class HighPriorityCusEmployee {

    @FXML
    TextArea Display;
    @FXML
    TextArea Display1;
    @FXML
    TextArea Display2;
    @FXML
    TextField TRP1;
    @FXML
    TextField TRP2;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToScene2(ActionEvent event) throws IOException {
        root =
FXMLLoader.load(getClass().getResource("InnerCircleCusEmployee.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public void ShowApplicationNumbers() {

        List<String> HighPriorityCustomerApplicationNumbers =
Collections.emptyList();

        try {
            HighPriorityCustomerApplicationNumbers =
Files.readAllLines(Paths.get("C:\\Users\\HP\\Data Structure
```



```
CW\\HighPriorityCustomerAppNos.txt"), StandardCharsets.UTF_8);
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Required file is not found");
        }

        Display.clear();

        for (int i = 0; i < HighPriorityCustomerApplicationNumbers.size();
i++) {
            Display.appendText(HighPriorityCustomerApplicationNumbers.get(i)
+ "\n");
        }
    }

    public void ApproveProcess() {
        ArrayList<String> arrayApprove1 = new ArrayList<>();

        arrayApprove1.add(TRP1.getText());

        for (int i = 0; i < arrayApprove1.size(); i++) {
            Display1.appendText(arrayApprove1.get(i) + "\n");
        }
    }

    public void DispproveProcess() {
        ArrayList<String> arrayDisapprove = new ArrayList<>();

        arrayDisapprove.add(TRP2.getText());

        for (int i = 0; i < arrayDisapprove.size(); i++) {
            Display2.appendText(arrayDisapprove.get(i) + "\n");
        }
    }
}
```

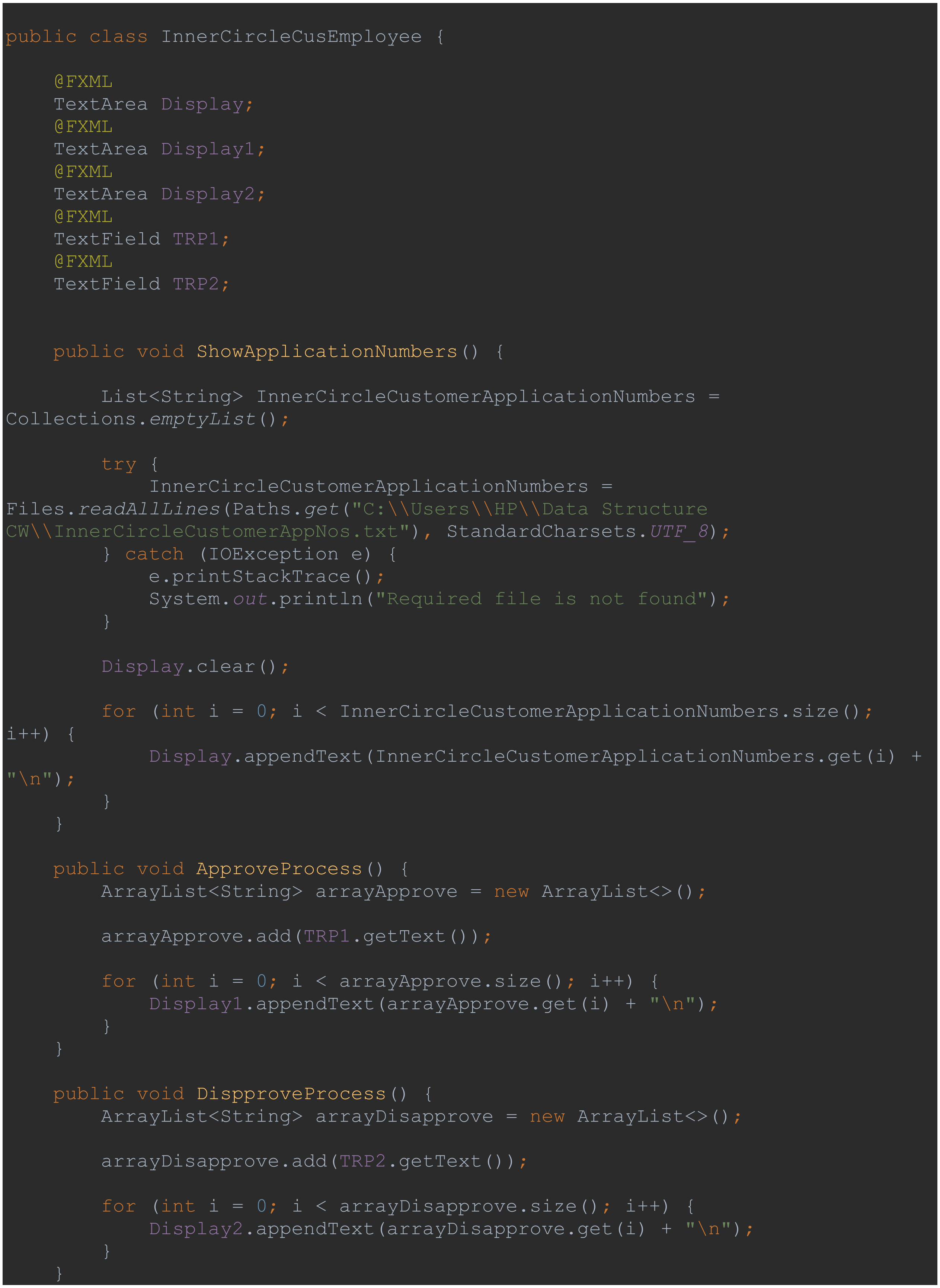
HighPriorityCustomer Class:

```
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class HighPriorityCustomer {
    Queue HighPriorityCustomerQueue = new Queue();
    ArrayList<String> arrayProcess1 = new ArrayList<>();
    String appNo;

    @FXML
    TextField TRP1;
    @FXML
    TextField TRP2;
    @FXML
    TextField TRP3;
    @FXML
    TextField TRP4;
    @FXML
    TextField TRP5;
    @FXML
    TextField TRP6;
    @FXML
    TextArea text1;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToScene2(ActionEvent event) throws IOException {
        root =
FXMLLoader.load(getClass().getResource("InnerCircleCustomer.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public void AddToQueue() {

        arrayProcess1.add(TRP1.getText());
        HighPriorityCustomerQueue.enqueue(Integer.parseInt(TRP1.getText()),
TRP2.getText(), Integer.parseInt(TRP3.getText()), TRP4.getText(),
TRP5.getText(), TRP6.getText());
```

```
    }

    public void ShowApplications() throws IOException {
        //System.out.println("Queue Front : " +
HighPriorityCustomerQueue.front.applicationNumber);
        //System.out.println("Queue Rear  : " +
HighPriorityCustomerQueue.rear.applicationNumber);
        text1. setText("Queue Front : " +
String.valueOf(HighPriorityCustomerQueue.front.applicationNumber) + " " +
HighPriorityCustomerQueue.front.NIC + " " +
HighPriorityCustomerQueue.front.accountNumber + " " +
HighPriorityCustomerQueue.front.collateralDescription + " " +
HighPriorityCustomerQueue.front.loanType + " " +
HighPriorityCustomerQueue.front.reasonOfLoan + "\n");
        text1.appendText("Queue Rear  : " +
String.valueOf(HighPriorityCustomerQueue.rear.applicationNumber) + " " +
HighPriorityCustomerQueue.rear.NIC + " " +
HighPriorityCustomerQueue.rear.accountNumber + " " +
HighPriorityCustomerQueue.rear.collateralDescription + " " +
HighPriorityCustomerQueue.rear.loanType + " " +
HighPriorityCustomerQueue.rear.reasonOfLoan + "\n");
        System.out.println(arrayProcess1);


        FileWriter writer = new FileWriter("HighPriorityCustomerAppNos.txt");
        for(String str: arrayProcess1) {
            writer.write(str + System.lineSeparator());
        }
        writer.close();

    }
}
```

InnerCircleCusEmployee Class:

```
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.List;
```



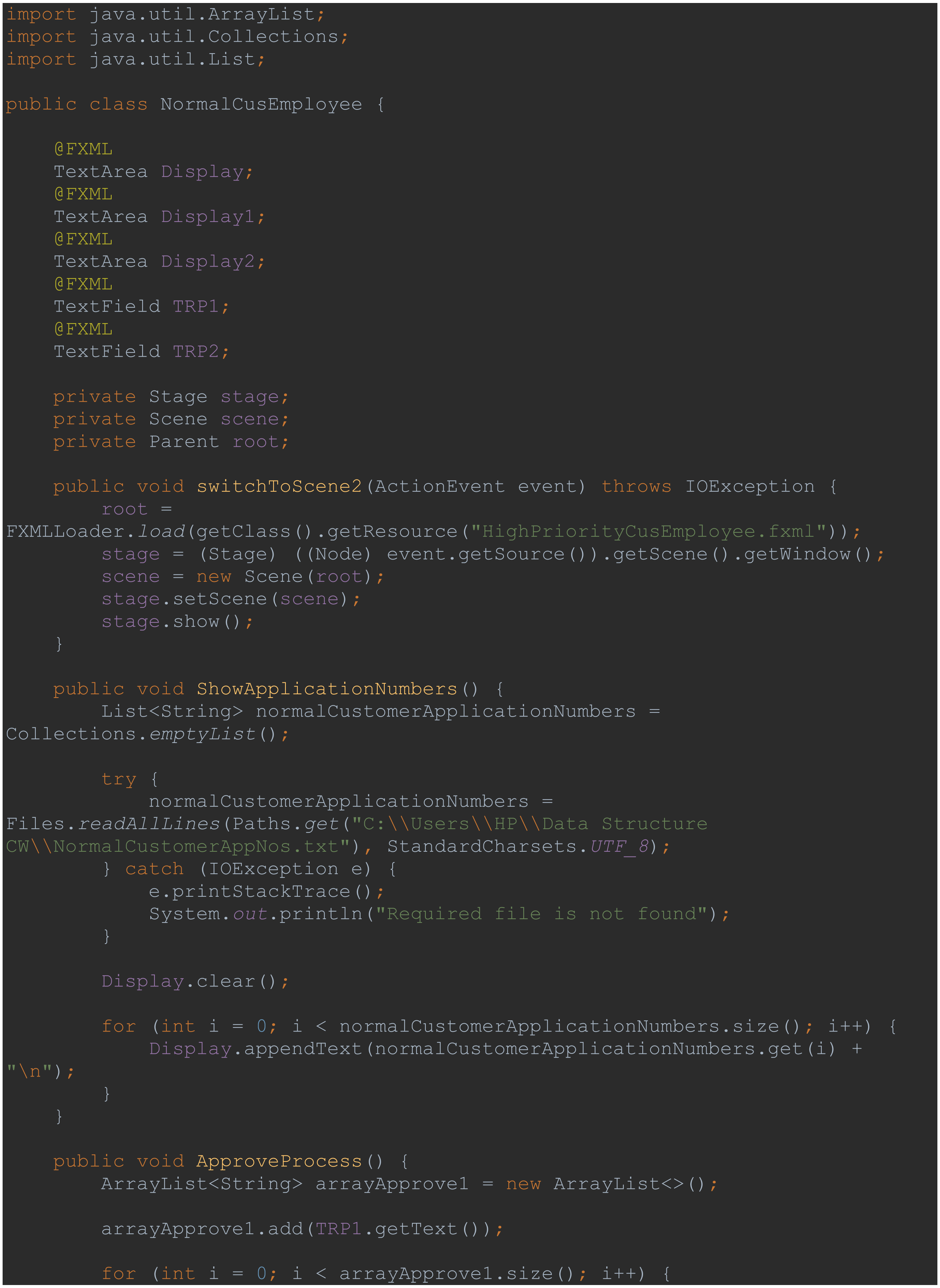
```
public class InnerCircleCusEmployee {

    @FXML
    TextArea Display;
    @FXML
    TextArea Display1;
    @FXML
    TextArea Display2;
    @FXML
    TextField TRP1;
    @FXML
    TextField TRP2;


    public void ShowApplicationNumbers() {

        List<String> InnerCircleCustomerApplicationNumbers =
Collections.emptyList();

        try {
            InnerCircleCustomerApplicationNumbers =
Files.readAllLines(Paths.get("C:\\Users\\HP\\Data Structure
CW\\InnerCircleCustomerAppNos.txt"), StandardCharsets.UTF_8);
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Required file is not found");
        }

        Display.clear();

        for (int i = 0; i < InnerCircleCustomerApplicationNumbers.size();
i++) {
            Display.appendText(InnerCircleCustomerApplicationNumbers.get(i) +
"\n");
        }
    }

    public void ApproveProcess() {
        ArrayList<String> arrayApprove = new ArrayList<>();

        arrayApprove.add(TRP1.getText());

        for (int i = 0; i < arrayApprove.size(); i++) {
            Display1.appendText(arrayApprove.get(i) + "\n");
        }
    }

    public void DispproveProcess() {
        ArrayList<String> arrayDisapprove = new ArrayList<>();

        arrayDisapprove.add(TRP2.getText());

        for (int i = 0; i < arrayDisapprove.size(); i++) {
            Display2.appendText(arrayDisapprove.get(i) + "\n");
        }
    }
}
```

InnerCircleCustomer Class:

```
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class InnerCircleCustomer {
    Queue InnerCircleCustomerQueue = new Queue();
    ArrayList<String> arrayProcess2 = new ArrayList<>();
    String appNo;

    @FXML
    TextField TRP1;
    @FXML
    TextField TRP2;
    @FXML
    TextField TRP3;
    @FXML
    TextField TRP4;
    @FXML
    TextField TRP5;
    @FXML
    TextField TRP6;
    @FXML
    TextArea text2;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToScene2(ActionEvent event) throws IOException {
        root =
FXMLLoader.load(getClass().getResource("NormalCusEmployee.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public void AddToQueue() {
        arrayProcess2.add(TRP1.getText());
        InnerCircleCustomerQueue.enqueue(Integer.parseInt(TRP1.getText()),
TRP2.getText(), Integer.parseInt(TRP3.getText()), TRP4.getText(),
TRP5.getText(), TRP6.getText());

    }

    public void ShowApplications() throws IOException {
        //System.out.println("Queue Front : " +
InnerCircleCustomerQueue.front.applicationNumber);
        //System.out.println("Queue Rear  : " +
InnerCircleCustomerQueue.rear.applicationNumber);
        text2. setText(String.valueOf("Queue Front : " +
InnerCircleCustomerQueue.front.applicationNumber) + " " +
InnerCircleCustomerQueue.front.NIC + " " +
InnerCircleCustomerQueue.front.accountNumber + " " +
InnerCircleCustomerQueue.front.collateralDescription + " " +
InnerCircleCustomerQueue.front.loanType + " " +
InnerCircleCustomerQueue.front.reasonOfLoan + "\n");
        text2.appendText(String.valueOf("Queue Rear  : " +
InnerCircleCustomerQueue.rear.applicationNumber) + " " +
InnerCircleCustomerQueue.rear.NIC + " " +
InnerCircleCustomerQueue.rear.accountNumber + " " +
InnerCircleCustomerQueue.rear.collateralDescription + " " +
InnerCircleCustomerQueue.rear.loanType + " " +
InnerCircleCustomerQueue.rear.reasonOfLoan + "\n");
        System.out.println(arrayProcess2);

        FileWriter writer = new FileWriter("InnerCircleCustomerAppNos.txt");
        for(String str: arrayProcess2) {
            writer.write(str + System.lineSeparator());
        }
        writer.close();

    }
}
```

NormalCusEmployee Class:

```
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
```



```
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class NormalCusEmployee {

    @FXML
    TextArea Display;
    @FXML
    TextArea Display1;
    @FXML
    TextArea Display2;
    @FXML
    TextField TRP1;
    @FXML
    TextField TRP2;

    private Stage stage;
    private Scene scene;
    private Parent root;

    public void switchToScene2(ActionEvent event) throws IOException {
        root =
FXMLLoader.load(getClass().getResource("HighPriorityCusEmployee.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public void ShowApplicationNumbers() {
        List<String> normalCustomerApplicationNumbers =
Collections.emptyList();

        try {
            normalCustomerApplicationNumbers =
Files.readAllLines(Paths.get("C:\\Users\\HP\\Data Structure
CW\\NormalCustomerAppNos.txt"), StandardCharsets.UTF_8);
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Required file is not found");
        }

        Display.clear();

        for (int i = 0; i < normalCustomerApplicationNumbers.size(); i++) {
            Display.appendText(normalCustomerApplicationNumbers.get(i) +
"\n");
        }
    }

    public void ApproveProcess() {
        ArrayList<String> arrayApprove1 = new ArrayList<>();

        arrayApprove1.add(TRP1.getText());

        for (int i = 0; i < arrayApprove1.size(); i++) {
            Display1.appendText(arrayApprove1.get(i) + "\n");
        }
    }

    public void DispproveProcess() {
        ArrayList<String> arrayDisapprove = new ArrayList<>();

        arrayDisapprove.add(TRP2.getText());

        for (int i = 0; i < arrayDisapprove.size(); i++) {
            Display2.appendText(arrayDisapprove.get(i) + "\n");
        }
    }
}
```

NormalCustomer Class:

```
package sample;

import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.Node;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.scene.control.TextArea;
import javafx.scene.control.TextField;
import javafx.stage.Stage;

import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;

public class NormalCustomer {
    Queue NormalCustomerQueue = new Queue();
    ArrayList<String> arrayProcess = new ArrayList<>();
    String appNo;

    @FXML
    TextField TRP1;
    @FXML
    TextField TRP2;
    @FXML
    TextField TRP3;
    @FXML
    TextField TRP4;
    @FXML
    TextField TRP5;
    @FXML
    TextField TRP6;
    @FXML
    TextArea text3;

    private Stage stage;
    private Scene scene;
```

```java
    private Parent root;

    public void switchToScene2(ActionEvent event) throws IOException {
        root =
FXMLLoader.load(getClass().getResource("HighPriorityCustomer.fxml"));
        stage = (Stage) ((Node) event.getSource()).getScene().getWindow();
        scene = new Scene(root);
        stage.setScene(scene);
        stage.show();
    }

    public void AddToQueue() {

        arrayProcess.add(TRP1.getText());
        NormalCustomerQueue.enqueue(Integer.parseInt(TRP1.getText()),
TRP2.getText(), Integer.parseInt(TRP3.getText()), TRP4.getText(),
TRP5.getText(), TRP6.getText());

    }

    public void ShowApplications() throws IOException {
        //System.out.println("Queue Front : " +
NormalCustomerQueue.front.applicationNumber);
        //System.out.println("Queue Rear  : " +
NormalCustomerQueue.rear.applicationNumber);
        text3. setText("Queue Front : " +
String.valueOf(NormalCustomerQueue.front.applicationNumber) + " " +
NormalCustomerQueue.front.NIC + " " + NormalCustomerQueue.front.accountNumber
+ " " + NormalCustomerQueue.front.collateralDescription + " " +
NormalCustomerQueue.front.loanType + " " +
NormalCustomerQueue.front.reasonOfLoan + "\n");
        text3. appendText("Queue Rear  : " +
String.valueOf(NormalCustomerQueue.rear.applicationNumber) + " " +
NormalCustomerQueue.rear.NIC + " " + NormalCustomerQueue.rear.accountNumber +
" " + NormalCustomerQueue.rear.collateralDescription + " " +
NormalCustomerQueue.rear.loanType + " " +
NormalCustomerQueue.rear.reasonOfLoan + "\n");
        System.out.println(arrayProcess);


        FileWriter writer = new FileWriter("NormalCustomerAppNos.txt");
        for(String str: arrayProcess) {
            writer.write(str + System.lineSeparator());
        }
        writer.close();

    }
}
```

Task 2

(a)

Fibonacci Numbers – Fibonacci sequence is the series of numbers where each number is the sum of the two preceding numbers.

Example:

0,1,1,2,3,5,8,13,21,34,55,89,144,233,377, 610, ……

(b)

Applications of Fibonacci numbers:

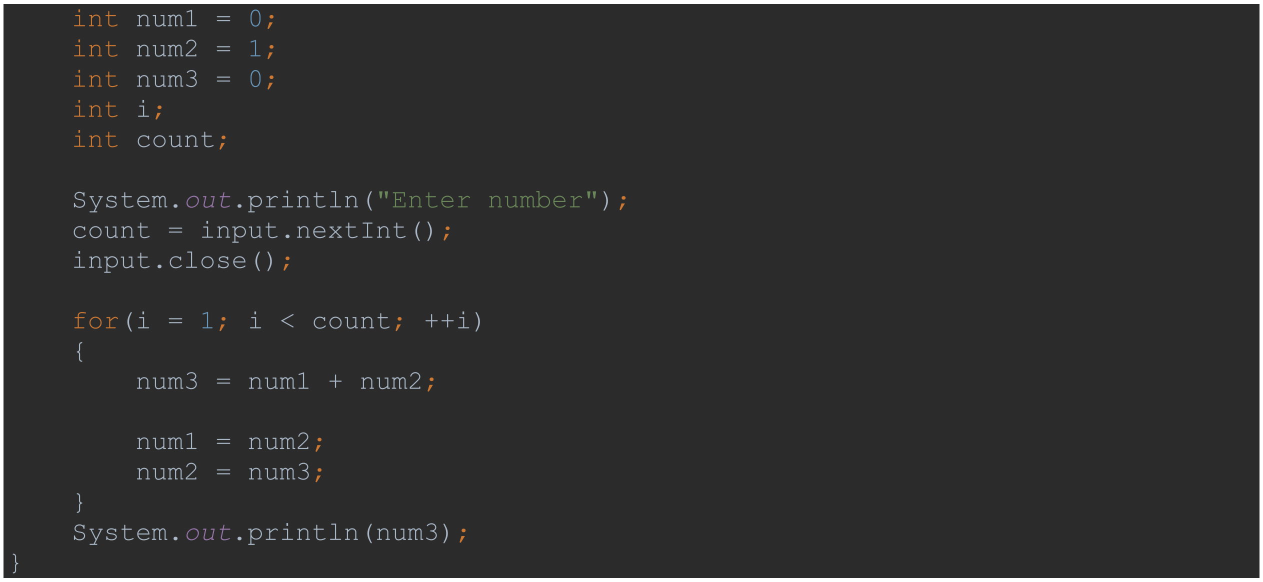
Fibonacci sequences are used in computer algorithms such as Fibonacci search technique, Fibonacci heap data structure and graphs also known as Fibonacci cubes used for interconnecting parallel and distributed systems.

Fibonacci heaps are used for priority queue operations.

Fibonacci retracements are used in predicting prices of stocks.

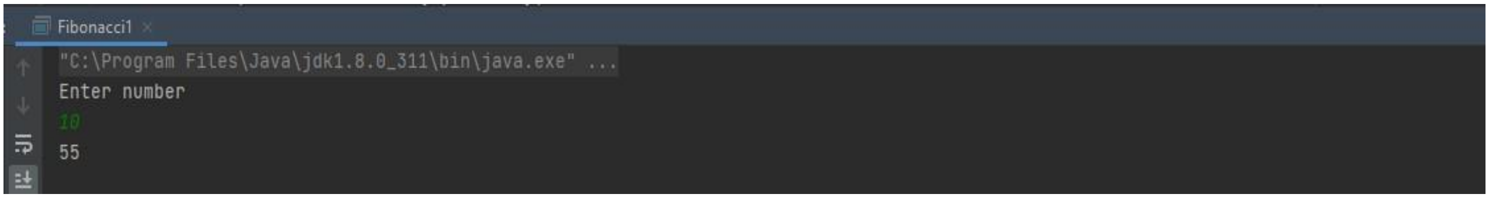
Fibonacci sequences can also be seen in biological settings such as arrangement of leaves in a stem, branching of a tree, fruit sprouts of a pineapple, the flowering of an artichoke, arrangement of pine cone’s bracts and an uncurling fern.

(c)



```
    int num1 = 0;
    int num2 = 1;
    int num3 = 0;
    int i;
    int count;

    System.out.println("Enter number");
    count = input.nextInt();
    input.close();

    for(i = 1; i < count; ++i)
    {
        num3 = num1 + num2;

        num1 = num2;
        num2 = num3;
    }
    System.out.println(num3);
}
```

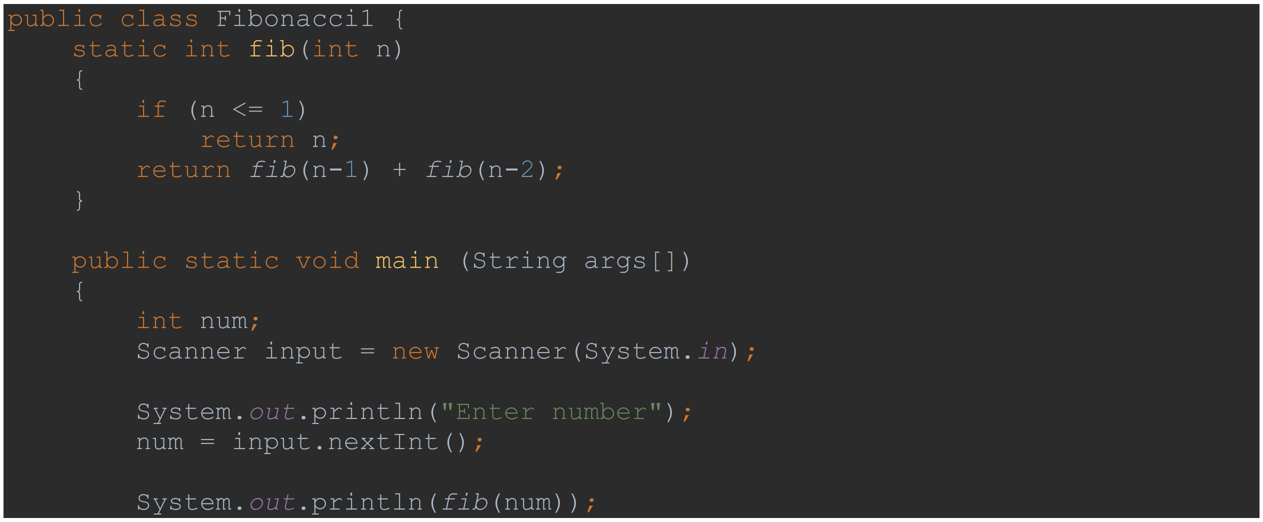
OUTPUT:



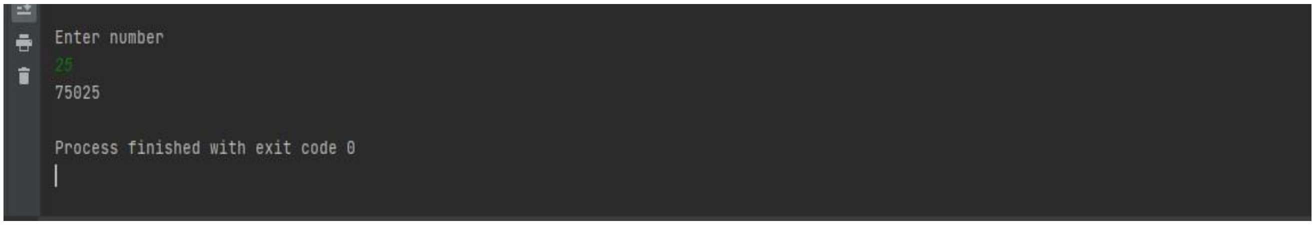
```
"C:\Program Files\Java\jdk1.8.0_311\bin\java.exe" ...
Enter number
10
55
```

Test No	Input	Expected Output	Actual Output
01	5	5	5
02	10	55	55
03	25	75025	75025
04	30	832040	832040
05	40	102334155	102334155

(d)



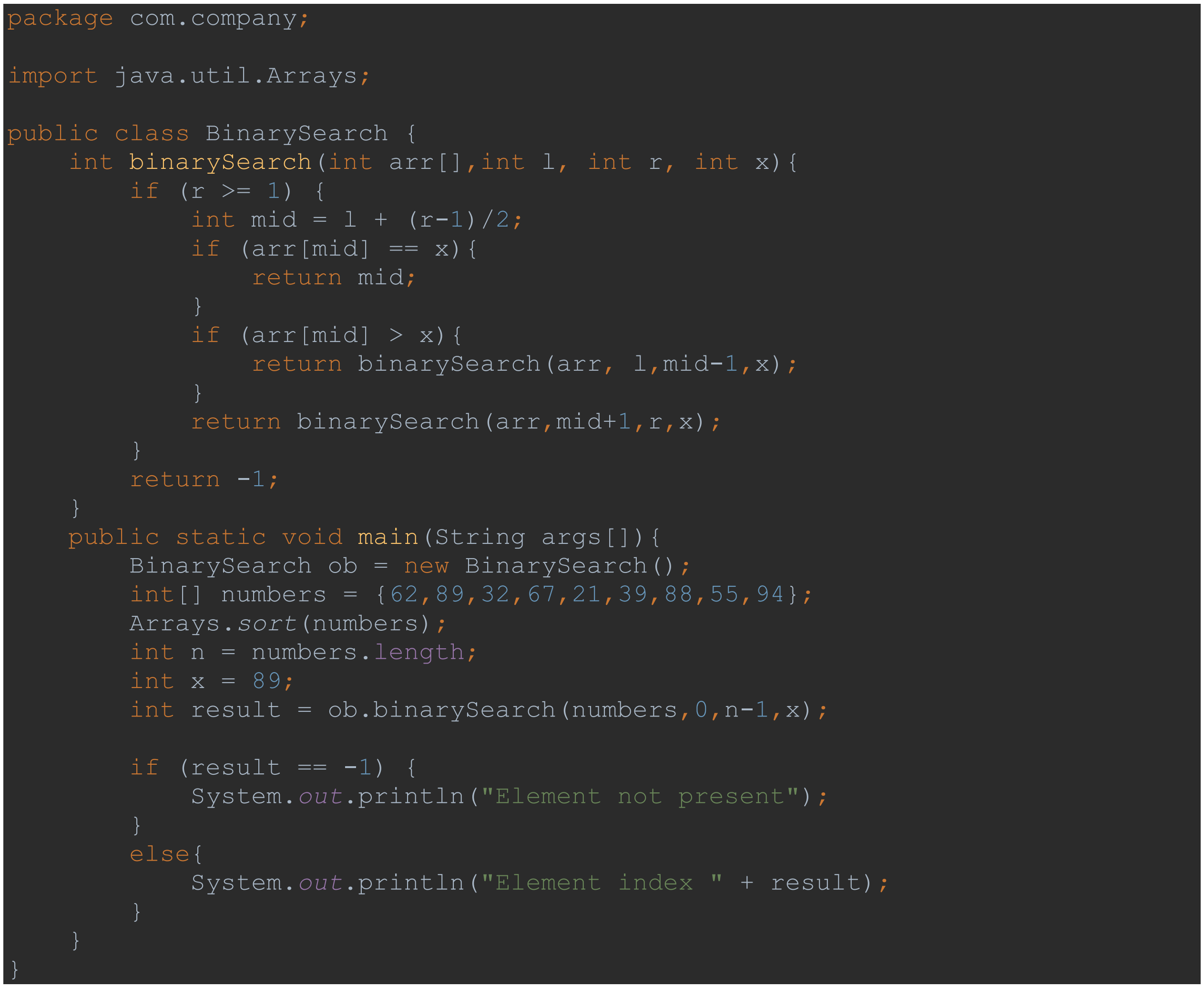
```
public class Fibonacci1 {
    static int fib(int n)
    {
        if (n <= 1)
            return n;
        return fib(n-1) + fib(n-2);
    }

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

        System.out.println("Enter number");
        num = input.nextInt();

        System.out.println(fib(num));
```



```
Enter number
25
75025

Process finished with exit code 0
```

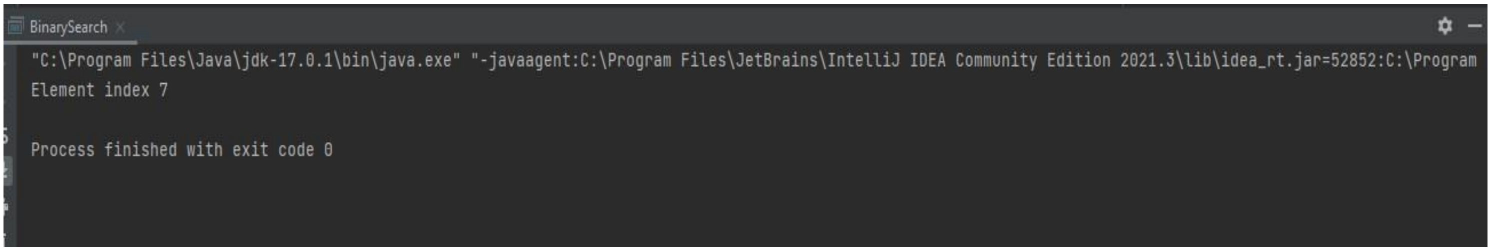
Test No	Input	Expected Output	Actual Output
01	6	8	8
02	12	144	144
03	27	196418	196418
04	35	9227465	9227465
05	45	1134903170	1134903170

(e)**BINARY SEARCH**



```java
package com.company;

import java.util.Arrays;

public class BinarySearch {
    int binarySearch(int arr[],int l, int r, int x){
        if (r >= 1) {
            int mid = l + (r-1)/2;
            if (arr[mid] == x){
                return mid;
            }
            if (arr[mid] > x){
                return binarySearch(arr, l,mid-1,x);
            }
            return binarySearch(arr,mid+1,r,x);
        }
        return -1;
    }
    public static void main(String args[]){
        BinarySearch ob = new BinarySearch();
        int[] numbers = {62,89,32,67,21,39,88,55,94};
        Arrays.sort(numbers);
        int n = numbers.length;
        int x = 89;
        int result = ob.binarySearch(numbers,0,n-1,x);

        if (result == -1) {
            System.out.println("Element not present");
        }
        else{
            System.out.println("Element index " + result);
        }
    }
}
```

OUTPUT:

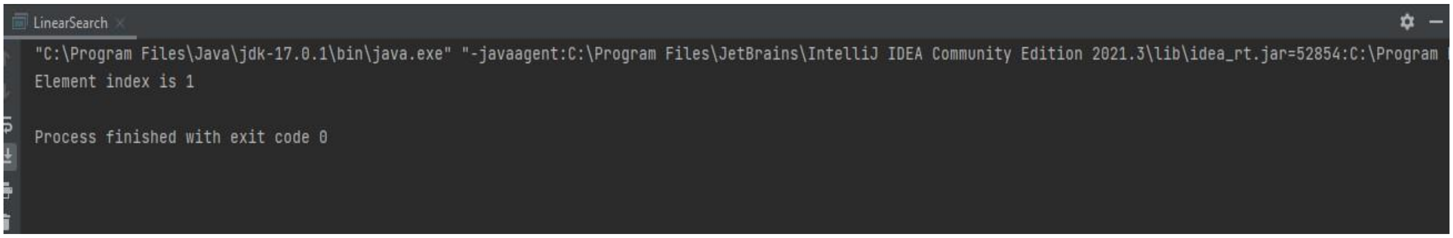


```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3\lib\idea_rt.jar=52852:C:\Program
Element index 7

Process finished with exit code 0
```

LINEAR SEARCH

```java
package com.company;

public class LinearSearch {
    public static int search(int arr[],int x){
        int n = arr.length;
        for (int i = 0; i<n; i++) {
            if (arr[i] == x)
                return i;
        }
        return -1;
    }
    public static void main(String args[]) {
        int arr[] = {62,89,32,67,21,39,88,55,94};
        int x= 89;

        int result = search(arr,x);
        if (result == -1) {
            System.out.println("Element not present");
        }
        else {
            System.out.println("Element index is " + result);
        }
    }
}
```

OUTPUT:

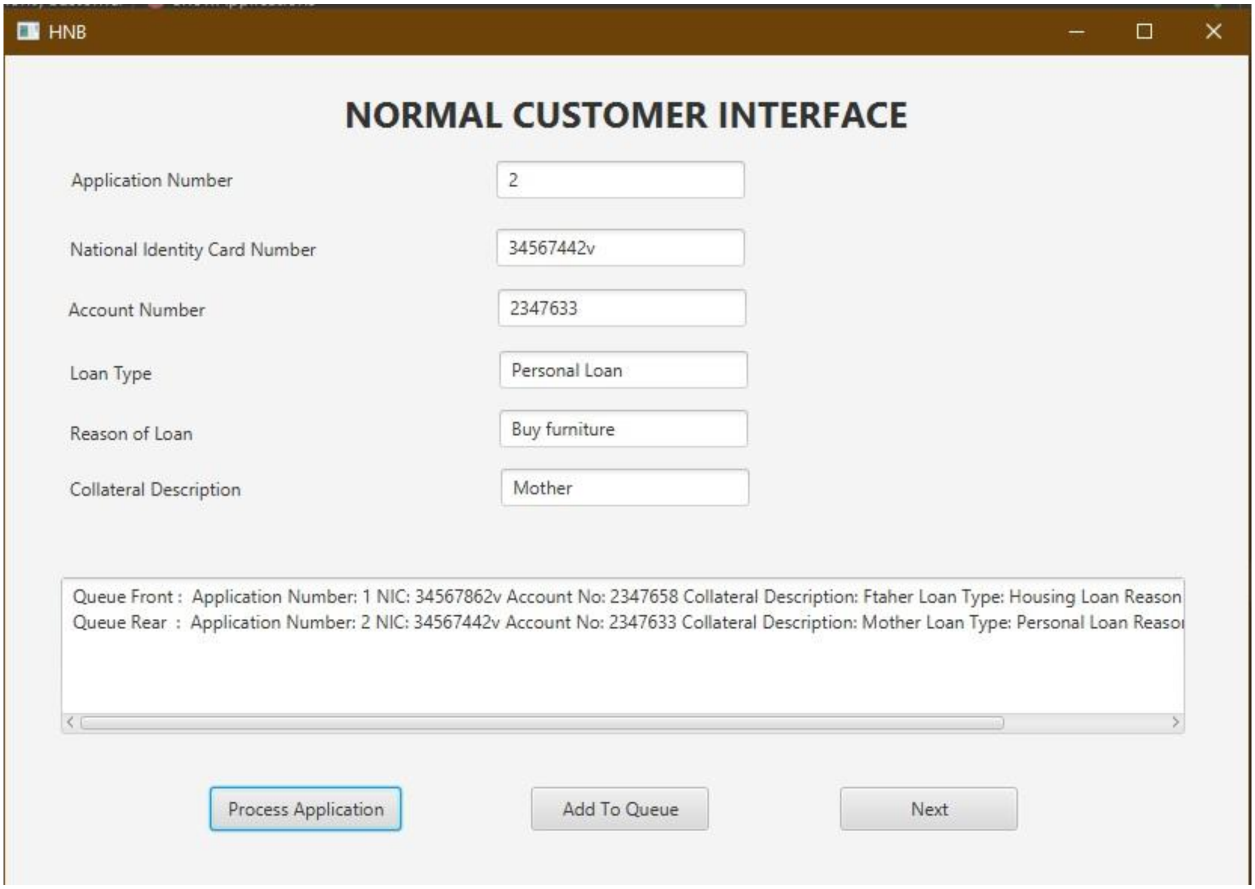


```
"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.3\lib\idea_rt.jar=52854:C:\Program
Element index is 1

Process finished with exit code 0
```

(f)

Linear Search	**Binary Search**
Starts searching from the first element and it compares each element with the searched element till the element isn’t found.	Finds the position of the searched element by finding the middle element of an array.
It is not necessary to arrange the elements in sorted order.	It is mandatory to arrange the elements in sorted order.
Can be implemented on any linear data structure such as an array or a linked list.	Can be implemented only on those data structures that have two-way traversal.
Based on the sequential approach.	Based on the divide and conquer approach.
This is preferrable for the small-sized data sets.	This is preferrable for the large-size data sets.
Less efficient in the case of large-size data sets.	More efficient in the case of large-size data sets.
O(n) is considered as the worst- case scenario for finding the element.	O(log2n) is considered as the worst- case scenario for finding the element.
O(1) is considered as the best-case scenario for finding the first element in the list.	O(1) is considered as the best-case scenario for finding the first element in the list.
Implementation can be done on both single and multidimensional array.	Implementation can be done only on a multidimensional array.

Screenshots of the GUI:



NORMAL CUSTOMER INTERFACE

Application Number: 2

National Identity Card Number: 34567442v

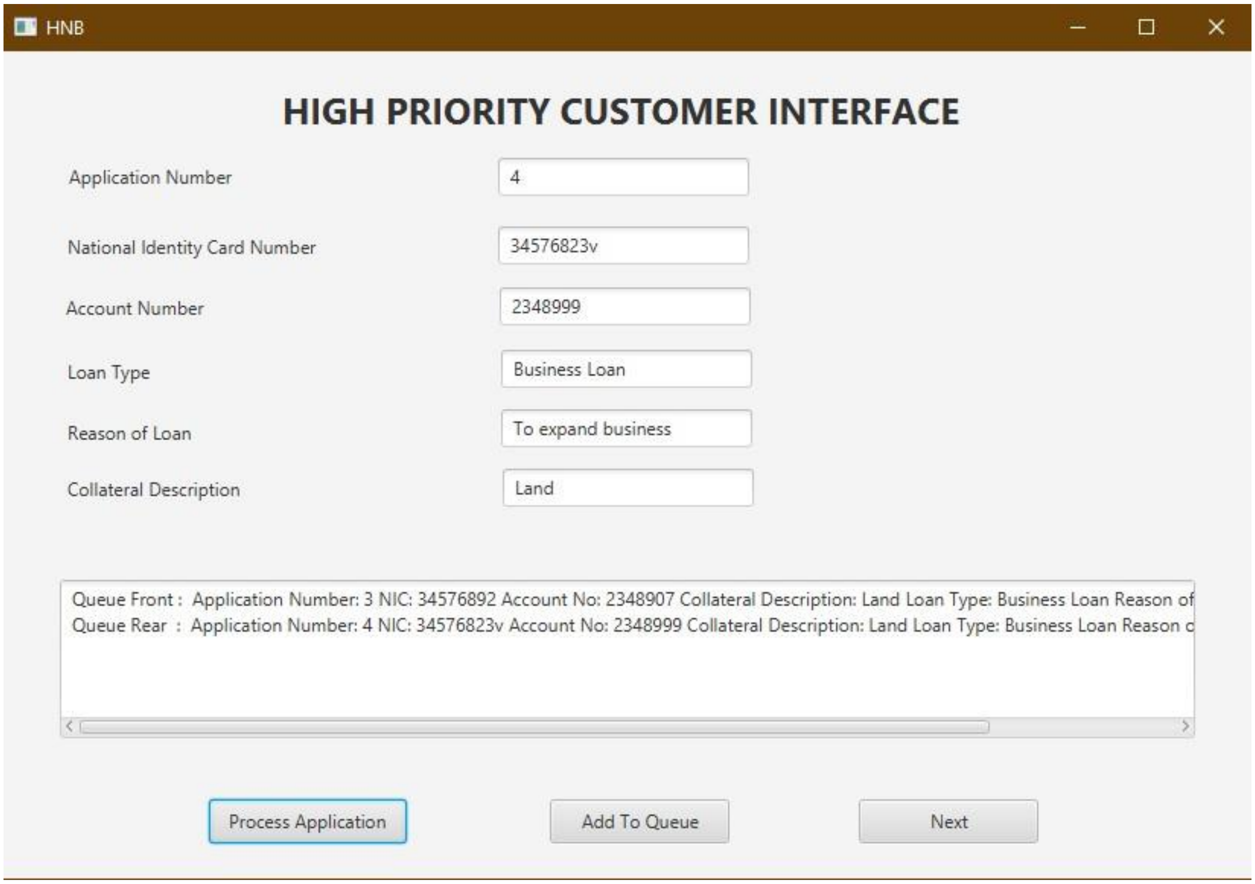
Account Number: 2347633

Loan Type: Personal Loan

Reason of Loan: Buy furniture

Collateral Description: Mother

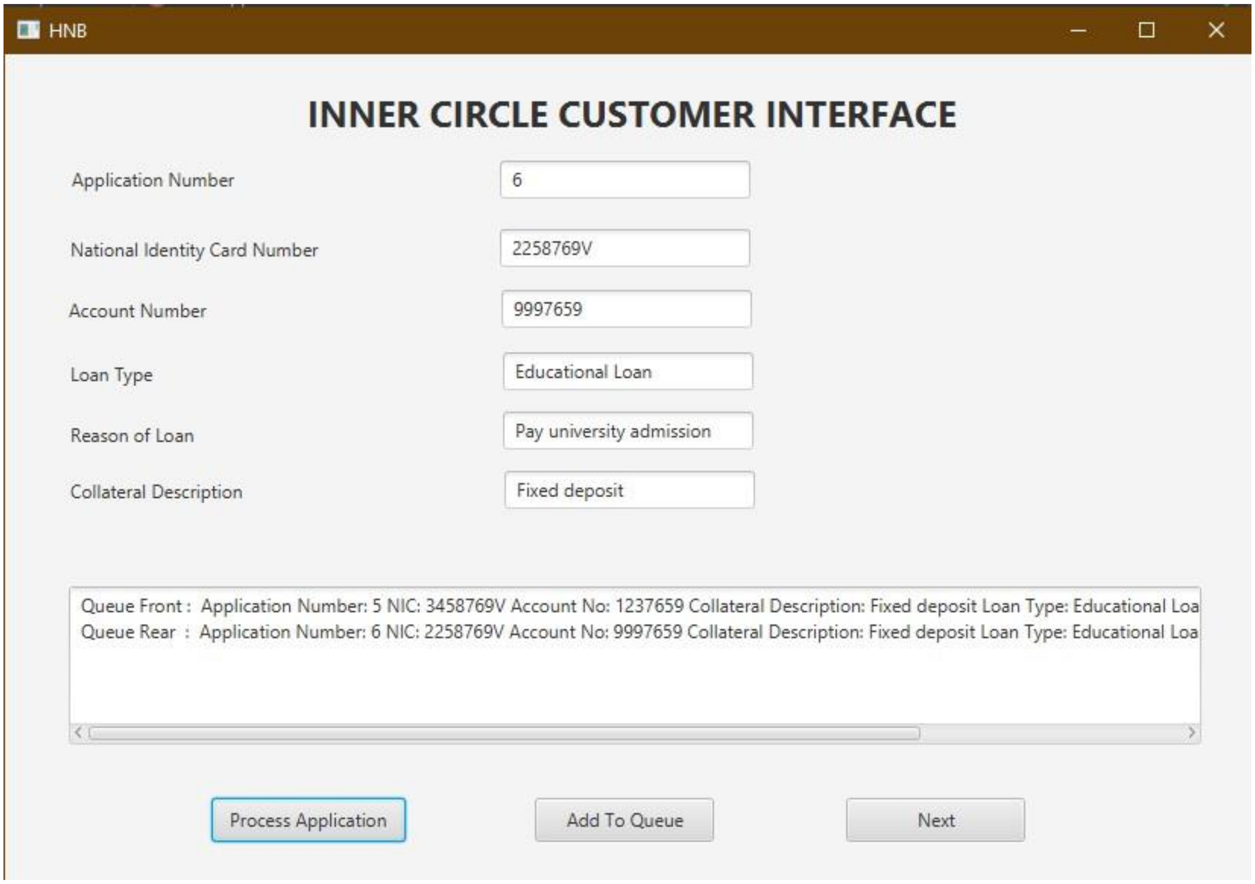
```
Queue Front :  Application Number: 1 NIC: 34567862v Account No: 2347658 Collateral Description: Ftaher Loan Type: Housing Loan Reason
Queue Rear  :  Application Number: 2 NIC: 34567442v Account No: 2347633 Collateral Description: Mother Loan Type: Personal Loan Reaso
```



HIGH PRIORITY CUSTOMER INTERFACE

Application Number	4
National Identity Card Number	34576823v
Account Number	2348999
Loan Type	Business Loan
Reason of Loan	To expand business
Collateral Description	Land

```
Queue Front :  Application Number: 3 NIC: 34576892 Account No: 2348907 Collateral Description: Land Loan Type: Business Loan Reason of
Queue Rear  :  Application Number: 4 NIC: 34576823v Account No: 2348999 Collateral Description: Land Loan Type: Business Loan Reason c
```



INNER CIRCLE CUSTOMER INTERFACE

Application Number 6

National Identity Card Number 2258769V

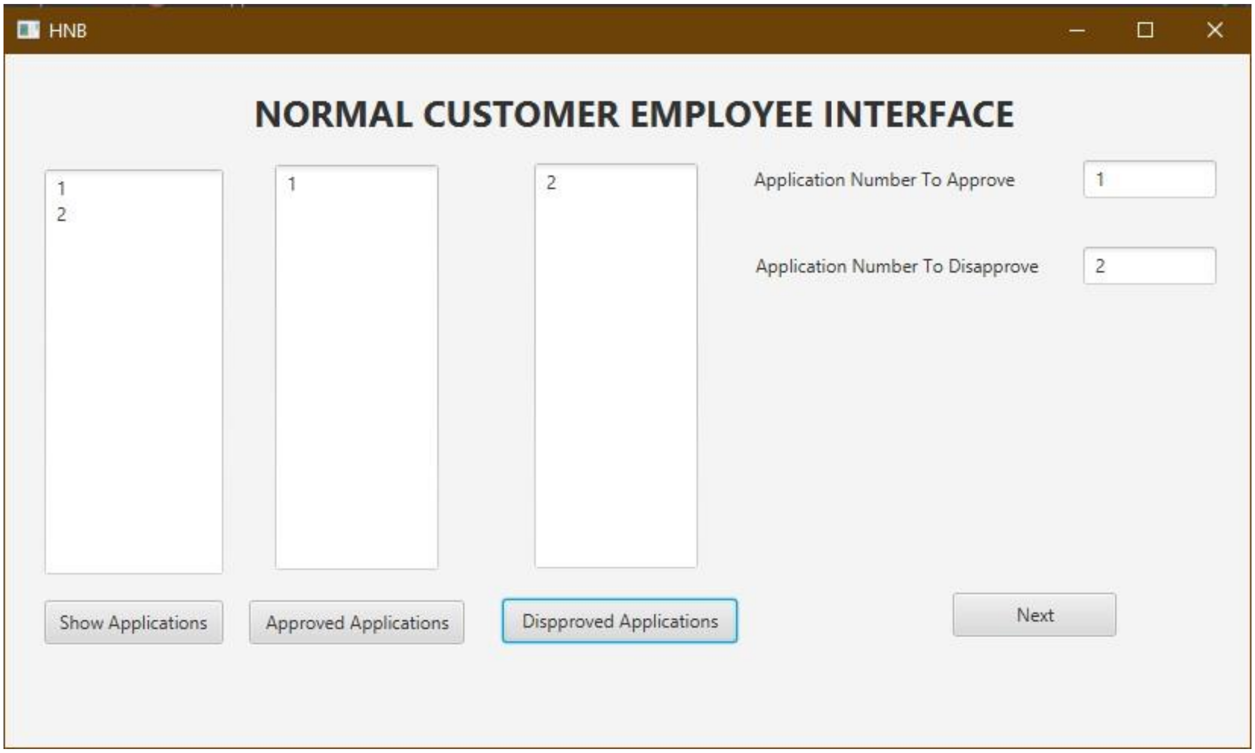
Account Number 9997659

Loan Type Educational Loan

Reason of Loan Pay university admission

Collateral Description Fixed deposit

```
Queue Front :  Application Number: 5 NIC: 3458769V Account No: 1237659 Collateral Description: Fixed deposit Loan Type: Educational Loa
Queue Rear  :  Application Number: 6 NIC: 2258769V Account No: 9997659 Collateral Description: Fixed deposit Loan Type: Educational Loa
```



NORMAL CUSTOMER EMPLOYEE INTERFACE

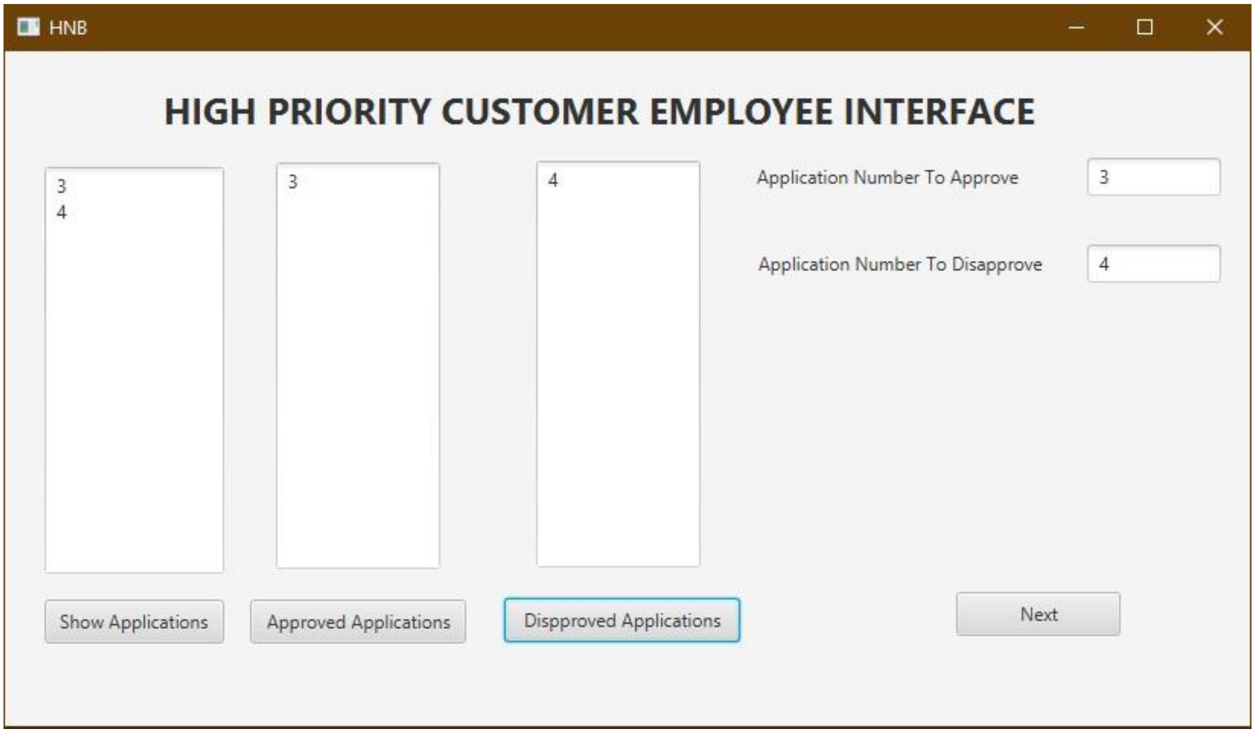
1
2

1

2

Application Number To Approve: 1

Application Number To Disapprove: 2



HIGH PRIORITY CUSTOMER EMPLOYEE INTERFACE

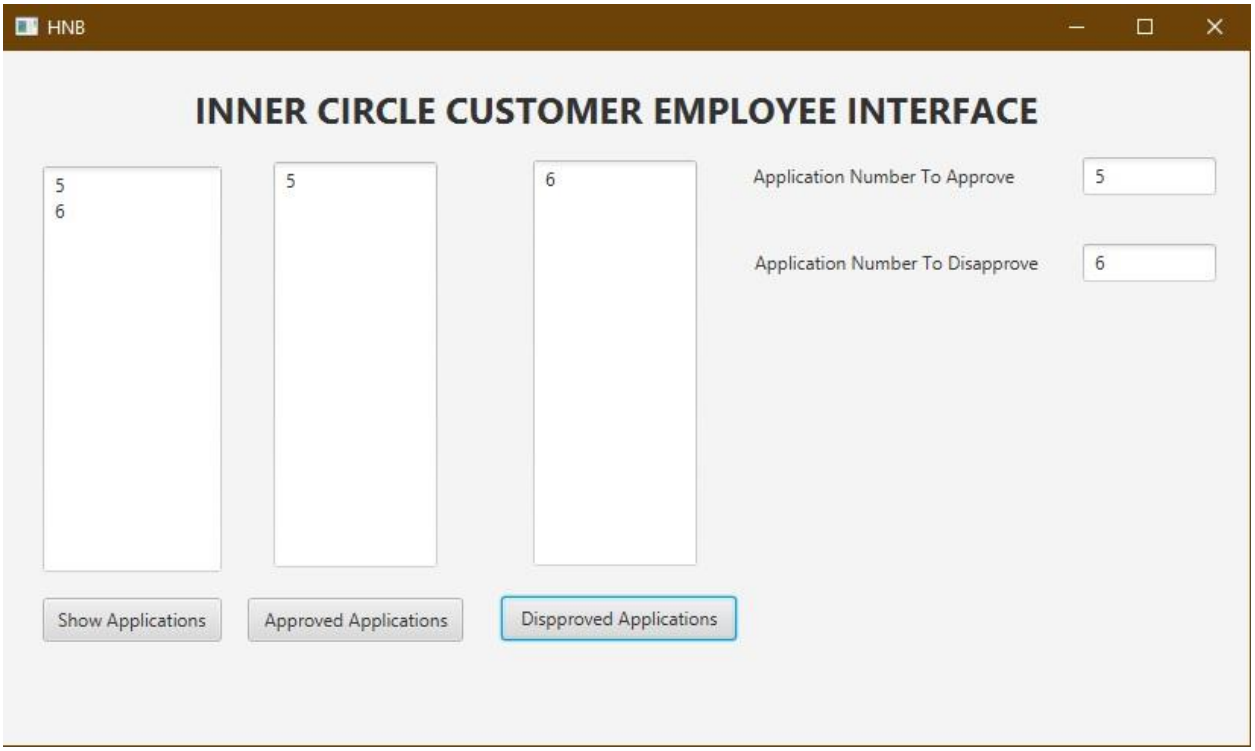
3
4

3

4

Application Number To Approve: 3

Application Number To Disapprove: 4



```
5
6
```

```
5
```

```
6
```

Application Number To Approve: 5

Application Number To Disapprove: 6