Assignment 4

Q.1 Write a java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second, and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.

Program:

```
import java.util.Random;

class RandomNumber extends Thread{

   public void run(){

      Random random=new Random();

      int count=0;

      while(count<=10){

         try{

            Thread.sleep(1000);// sleep for 1 second;

            int randomNumber = random.nextInt(10);

             if(randomNumber%2==0) {

               SquareThread squareThread = new SquareThread(randomNumber);

               squareThread.start();   }

            else{

               CubeThread cubeThread = new CubeThread(randomNumber);

               cubeThread.start();   }

         } catch(InterruptedException e){

            e.printStackTrace();   }

         count++;     } } }

class SquareThread extends Thread{

   private int number;

   public SquareThread(int number){

      this.number=number;   }

      public void run(){
```

```
      int square =number*number;
      System.out.println("square of "+number+" is : "+square);   }  }
class CubeThread extends Thread{
   private int number;
   public  CubeThread(int number){
      this.number=number; }
   public void run(){
      int cube =number*number*number;
      System.out.println("cube of "+number+" is : "+cube);   }    }
public class MultiThread{
   public static void main(String args[]){
      RandomNumber random = new RandomNumber();
      random.start();
  }   }
```

Output:

```
cube of 7 is : 343
cube of 5 is : 125
cube of 3 is : 27
square of 4 is : 16
cube of 1 is : 1
cube of 3 is : 27
square of 0 is : 0
cube of 5 is : 125
square of 6 is : 36
cube of 9 is : 729
```

Assignment 4

Q2.Write a java program that displays the number of characters, lines, and words in a text file.

Program:

```
import java.io.BufferedReader;

import java.io.FileReader;

import java.io.FileNotFoundException;

import java.io.IOException;

public class Filecount {

   static String path ="read.txt";

   public static void main(String args[]){

     int charcount=0;

      int wordcount=0;

      int linecount=0;

      BufferedReader  reader;

      try  {

         reader = new BufferedReader(new FileReader(path));

         String currline=reader.readLine();

          while(currline!=null)  {

            linecount++;

            //words

            String words[]=currline.split("  ");

            wordcount = wordcount+words.length;

           //char

            for(String word : words){

               charcount = charcount+word.length();    }

            currline= reader.readLine();   }

         System.out.println("total lines : "+linecount);
```

```
          System.out.println("total words : "+wordcount);
          System.out.println("total char : "+charcount);  }
        catch (FileNotFoundException e){
          e.printStackTrace();    }
        catch (IOException e)   {
          e.printStackTrace();  }
      } }
```

read.txt:

```
hey everyone this is imran here
your welcome in java world
let's start the journey
```

output:

```
total lines : 6
total words : 6
total char : 81
```

Q3.Write a java program that reads a file and displays the file on the screen with a line number before each line.

Program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;


public class DisplayNumber{
   public static void main(String args[]){
      String path="read.txt";
```

```
    try{
      BufferedReader reader = new BufferedReader(new FileReader(path));
      String line=reader.readLine();
      int lineNumber =1;
      while(line!=null){
        System.out.println(lineNumber+" : "+line);
        lineNumber++;
        line=reader.readLine();
      }
      reader.close();
    }catch(IOException e){
      e.printStackTrace(); }
  } }
```

Output:

```
1 : hey everyone this is imran here
2 : your welcome in java world
3 : let's start the journey
```

Q4.Write a java program that loads names and phone numbers from a text file where the data is organized as one line per record, and each field in record is separated by a tab (/t). It takes a name or phone number as input and prints the corresponding other value from the hash table. Hint: use hash tables.

Program:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.Hashtable;
```

Assignment 4

```
public class Phonebook {

  public static void main(String[] args) throws Exception {

    FileReader fileReader = new FileReader("demoText.txt");

    BufferedReader bufferedReader = new BufferedReader(fileReader);

    Hashtable<String, String> contact = new Hashtable<>();

    String line, name, number;

    while ((line = bufferedReader.readLine()) != null) {

      String words[] = line.split("\t");

      name = words[0];

      number = words[1];

      contact.put(name, number);

    }

    for (int i = 0; i < contact.size(); i++) {

      String namesArray[] = contact.keySet().toArray(new String[0]);

      System.out.println(namesArray[i] + " : " + contact.get(namesArray[i]));

    }

  } }
```

Output:

```
imran khan : 9875034945

bablu yadav : 4066520553

vishvash gaur : 9515945559

priya solanki : 7153959903

josheph stifan : 8137840988
```

Assignment 4

Q.5 Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.

Program:

```
import java.sql.*;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.PreparedStatement;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

public class NewJFrame extends javax.swing.JFrame {

Connection con;

ResultSet rs;

PreparedStatement pst;

public NewJFrame() {

      initComponents();

try{

         String url = "jdbc:oracle:thin://@localhost:1521:xe";

         Class.forName("oracle.jdbc.driver.OracleDriver");

         System.out.println("Driver loaded !");

         con= DriverManager.getConnection(url,"system","immu");

        System.out.println("connection  estabilished ");

         }

catch(ClassNotFoundException e){

         System.out.println("driver is not loaded");

      }

       catch(SQLException e){   System.out.println("connection not estabilished ");   }  }
```

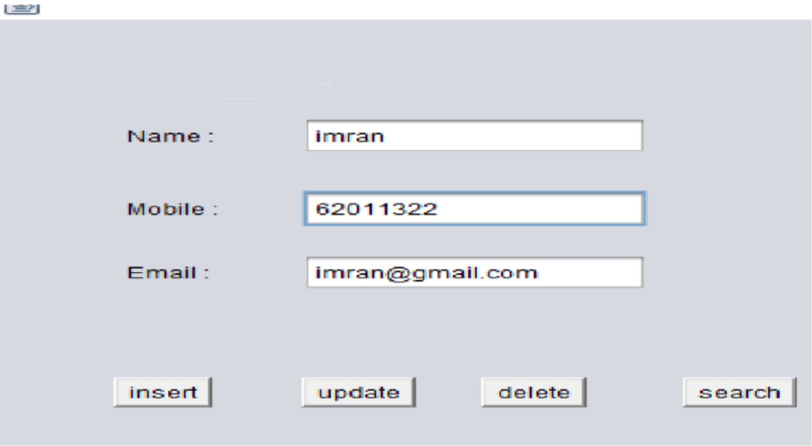
```
private void insertActionPerformed(java.awt.event.ActionEvent evt) {
   try {
      pst = con.prepareStatement("insert into students(name,mobile,email)values(?,?,?)");
      pst.setString(1,name.getText());
       pst.setString(2,mobile.getText());
       pst.setString(3,email.getText());
       pst.executeUpdate();
       JOptionPane.showMessageDialog(this,"success");
       name.setText(" ");
       mobile.setText(" ");
       email.setText(" ");
   } catch (SQLException ex) {
       Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
}  }
private void updateActionPerformed(java.awt.event.ActionEvent evt) {
   try {
       pst=con.prepareStatement("update students set mobile=?,email=? where name=? ");
       pst.setString(1, mobile.getText());
       pst.setString(2, email.getText());
       pst.setString(3, name.getText());
       pst.executeUpdate();
       JOptionPane.showMessageDialog(this,"success");
       name.setText(" ");
       mobile.setText(" ");
       email.setText(" ");
       } catch (SQLException ex) {
       Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);  }   }
```

```
private void deleteActionPerformed(java.awt.event.ActionEvent evt) {
   try {
       pst=con.prepareStatement("delete from students where name = ?");
       pst.setString(1, name.getText());
       pst.executeUpdate();
       JOptionPane.showMessageDialog(this,"success");
       name.setText(" ");
       mobile.setText(" ");
       email.setText(" ");
} catch (SQLException ex) {
       Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
   }  }
private void searchActionPerformed(java.awt.event.ActionEvent evt) {
   try {
        String s = JOptionPane.showInputDialog(this,"enter name for search record : ");
       pst = con.prepareStatement(" select * from students where name=?");
       pst.setString(1,s);
       rs = pst.executeQuery();
       if(rs.next()){
       name.setText(rs.getString(1));
       mobile.setText(rs.getString(2));
       email.setText(rs.getString(3));
       }  } catch (SQLException ex) {
       Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
 }   }
public static void main(String args[]) {
             try {
```

```
        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }  }  } catch (ClassNotFoundException ex) {

    java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex)

     } catch (InstantiationException ex) {

    java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

      } catch (IllegalAccessException ex) {

    java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

      } catch (javax.swing.UnsupportedLookAndFeelException ex) {

    java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);

      }

      java.awt.EventQueue.invokeLater(new Runnable() {

          public void run() {

              new NewJFrame().setVisible(true);

          }  });}

private java.awt.Button delete;

    private javax.swing.JTextField email;

    private java.awt.Button insert;

    private javax.swing.JLabel labelemail;

   private javax.swing.JLabel labelmobile;

    private javax.swing.JLabel labelname;

    private javax.swing.JTextField mobile;
    private javax.swing.JTextField name;

    private java.awt.Button search;

    private java.awt.Button update;

    }
```

Output:

```
Driver loaded !

connection  estabilished
```

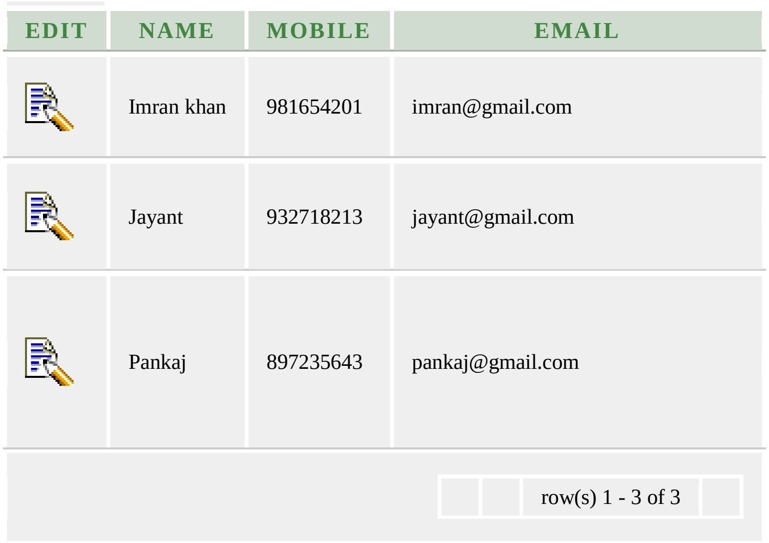


Name : imran

Mobile : 62011322

Email : imran@gmail.com

Database:



EDIT	NAME	MOBILE	EMAIL
✎	Imran khan	981654201	imran@gmail.com
✎	Jayant	932718213	jayant@gmail.com
✎	Pankaj	897235643	pankaj@gmail.com

row(s) 1 - 3 of 3