

1. Write a Java program to load and display multiple images into a swing container.

```
import java.awt.FlowLayout;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;

import javax.imageio.ImageIO;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class Image extends JFrame {
    private JLabel label1;
    private JLabel label2;
    Image() throws IOException{
        setLayout(new FlowLayout());

        //image1 = new ImageIcon(getClass().getResource("ICE_logo.jpg")); for jdk old version(8,6
        etc)

        File file = new
        File("D:\\Java program\\Pust_Session_2018_19\\src\\ICE_logo.jpg");

        BufferedImage image1 = ImageIO.read(file);

        ImageIcon imageIcon = new ImageIcon(image1);

        label1 = new JLabel(imageIcon);

        add(label1);

        //image2 = new ImageIcon(getClass().getResource("Pust_logo.png")); for jdk old version

        File file2 = new File("D:\\Java program\\Pust_Session_2018_19\\src\\Pust_logo.png");

        BufferedImage image2 = ImageIO.read(file2);

        ImageIcon imageIcon2 = new ImageIcon(image2);
```

```

label2 = new JLabel(imageIcon2);
add(label2);
}

public static void main(String args[]) throws IOException {
    Image gui = new Image();
    gui.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    gui.setVisible(true);
    gui.setSize(500,500);
    // gui.pack(); Fit image to the prescribe size of container
    gui.setTitle("Image Program");
}
}

```

2. Suppose a restaurant sells Pizza for \$100, Burgers for \$30, and Tea for \$10. Write a Java program for generating restaurant bills after ordering from a customer.

```

import javax.swing.*.*;
import java.awt.event.*;

public class BillGeneration extends JFrame implements ActionListener{
    JLabel l;
    JCheckBox cb1,cb2,cb3;
    JButton b;

    BillGeneration(){
        l=new JLabel("Food Ordering System");
        l.setBounds(50,50,300,20);
        cb1=new JCheckBox("Pizza @ 100");
        cb1.setBounds(100,100,150,20);
        cb2=new JCheckBox("Burger @ 30");
        cb2.setBounds(100,150,150,20);
    }
}

```

```

cb3=new JCheckBox("Tea @ 10");
cb3.setBounds(100,200,150,20);
b=new JButton("Order");
b.setBounds(100,250,80,30);
b.addActionListener(this);
add(l);add(cb1);add(cb2);add(cb3);add(b);
setSize(400,400);
setLayout(null);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);
}
public void actionPerformed(ActionEvent e){
float amount=0;
String msg="";
if(cb1.isSelected()){
amount+=100;
msg="Pizza: 100\n";
}
if(cb2.isSelected()){
amount+=30;
msg+="Burger: 30\n";
}
if(cb3.isSelected()){
amount+=10;
msg+="Tea: 10\n";
}
msg+="-----\n";
JOptionPane.showMessageDialog(this,msg+"Total: "+amount);

```

```
}  
  
public static void main(String[] args) {  
    new BillGeneration();  
}  
}
```

3. Write a Java Program to create a student registration form for ICE department including the fields “Name”, “Roll”, and “Department” in GUI.

```
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import javax.swing.JButton;  
import javax.swing.JFrame;  
import javax.swing.JLabel;  
import javax.swing.JPanel;  
import javax.swing.JTextField;  
  
public class Form implements ActionListener {  
    private static JLabel success;  
    private static JFrame frame;  
    private static JLabel label1,label2,label3;  
    private static JPanel panel;  
    private static JButton button;  
    private static JTextField userText1, userText2, userText3;  
  
    public static void main(String[] args) {  
        frame = new JFrame();  
        panel = new JPanel();  
        frame.setSize(400, 300);  
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        frame.add(panel);
```

```
panel.setLayout(null);
//Setting all Three Labels
label1= new JLabel("Name");
label1.setBounds(10,10,80,25);
panel.add(label1);label2 = new JLabel("Roll");
label2.setBounds(10,60,80,25);
panel.add(label2);
label3 = new JLabel("Department");
label3.setBounds(10,110,80,25);
panel.add(label3);
//Creating all Textfields
userText1 = new JTextField("Enter Your Name");
userText1.setBounds(100,10,200,25);
panel.add(userText1);
JTextField userText2 = new JTextField("Enter Your Roll");
userText2.setBounds(100,60,200,25);
panel.add(userText2);
JTextField userText3 = new JTextField("Enter Your Department");
userText3.setBounds(100,110,200,25);
panel.add(userText3);
button = new JButton("Save");
button.setBounds(150, 160, 80, 25);
button.addActionListener(new Form()); //not use this keyword,
//because it called from main method, if we use constructor then we can use this keyword
panel.add(button);
success = new JLabel("Not Submit Yet");
success.setBounds(130,210,300,25);
panel.add(success);
```

```

frame.setVisible(true);
}
@Override
public void actionPerformed(ActionEvent e) {
// TODO Auto-generated method stub
success.setText("Saved Successfully");
}
}

```

4. Write a Java Program in GUI to develop a simple calculator that can calculate addition, subtraction, division, and multiplication operations.

```

package new_java;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;

public class Calculator extends JFrame implements ActionListener
{
    JButton b10,b11,b12,b13,b14,b15;
    JButton b[]=new JButton[10];
    int i,r,n1,n2;
    JTextField res;
    char op;
    public Calculator()
    {
        super("Calulator");
        setLayout(new BorderLayout());
    }
}

```

```
JPanel p=new JPanel();
p.setLayout(new GridLayout(4,4));
for(int i=0;i<=9;i++)
{
b[i]=new JButton(i+"");
p.add(b[i]);
b[i].addActionListener(this);
}
b10=new JButton("+");
p.add(b10);
b10.addActionListener(this);
b11=new JButton("-");
p.add(b11);
b11.addActionListener(this);
b12=new JButton("*");
p.add(b12);
b12.addActionListener(this);
b13=new JButton("/");
p.add(b13);
b13.addActionListener(this);
b14=new JButton("=");
p.add(b14);
b14.addActionListener(this);
b15=new JButton("C");
p.add(b15);
b15.addActionListener(this);
res=new JTextField(10);
add(p, BorderLayout.CENTER);
```

```
add(res, BorderLayout.NORTH);
setVisible(true);
setSize(200,200);
}
public void actionPerformed(ActionEvent ae){
    JButton pb=(JButton)ae.getSource();
    if(pb==b15)
    {
        r=n1=n2=0;
        res.setText("");
    }
    else if(pb==b14)
    {
        n2=Integer.parseInt(res.getText());
        eval();
        res.setText(""+r);
    }
    else
    {
        boolean opf=false;
        if(pb==b10)
        { op='+';
          opf=true;
        }
        if(pb==b11)
        { op='-';opf=true;}
        if(pb==b12)
        { op='*';opf=true;}
```



```
if(pb==b13)
{ op='/';opf=true;}
if(opf==false)
{
for(i=0;i<10;i++)
{
if(pb==b[i])
{
String t=res.getText();
t+=i;
res.setText(t);
}
}
}
else
{
n1=Integer.parseInt(res.getText());
res.setText("");
}
}
}
int eval()
{
switch(op)
{
case '+': r=n1+n2; break;
case '-': r=n1-n2; break;
case '*': r=n1*n2; break;
```

```

case '/': r=n1/n2; break;
}
return 0;}

public static void main(String arg[])
{
new Calculator();
}
}

```

5. Suppose multiple threads try to access the same resources and finally produce erroneous and unforeseen results. Write a java program to solve this problem using object or method synchronization.

// File Name : Callme.java

```

class Callme {

    void call(String msg) {

        System.out.print "[" + msg);

        try {

            Thread.sleep(1000);

        } catch (InterruptedException e) {

            System.out.println("Interrupted"); }

        System.out.println("]");

    } }

```

// File Name : Caller.java

```

class Caller implements Runnable {

    String msg; Callme target; Thread t;

    public Caller(Callme targ, String s) {

        target = targ;
    }
}

```

```

        msg = s;
        t = new Thread(this);
        t.start(); }

// synchronize calls to call()
public void run() {
    //synchronized(target) // synchronized block
    {
        target.call(msg);
    }
} }

//File Name : Synch.java
class Synch {
    public static void main(String args[]) {
        Callme target = new Callme();
        Caller ob1 = new Caller(target, "Hello");
        Caller ob2 = new Caller(target, "Synchronized");
        Caller ob3 = new Caller(target, "World");

        // wait for threads to end
        try {
            ob1.t.join();
            ob2.t.join();
            ob3.t.join();
        } catch (InterruptedException e) {
            System.out.println("Interrupted");
        } } }

```

6. Write a client-server TCP socket program in java that the server listens for connection requests, and whatever message the client sends, the server converts it to uppercase and sends it back.

Client:

```
// SimpleClient.java: A simple client //program

import java.net.*;
import java.io.*;
import java.util.Scanner;

public class client1 {
    public static void main(String args[]) throws IOException {
        //Open your connection to a server, at      //port 2992
        Socket s1 = new Socket("localhost",2992);

        //Get an input file handle from the    //socket and read the input
        OutputStream s1out = s1.getOutputStream();
        DataOutputStream dos = new DataOutputStream (s1out);
        System.out.println("Write Your Message...");// take message from console
        Scanner sc=new Scanner(System.in);
        String msg=sc.nextLine();
        dos.writeUTF(msg); //transfer to server and turn into upper case
        //dos.writeUTF("hello world");
        InputStream s1In = s1.getInputStream();
        DataInputStream dis = new DataInputStream(s1In);
        String st = new String (dis.readUTF());
        System.out.println("Output:");
        System.out.println(st);
        //When done, just close the connection      //and exitdis.close();
        //s1In.close();
        //s1.close();
        //Send a string!
```

```

//dos.writeUTF("sent from client ");

//Close the connection, but not the
    //server socket

//dos.close();

//s1out.close();

s1.close();

}

}

```

Server:

```

//SimpleServer.java: A simple //server program

import java.net.*;
import java.io.*;

public class server1 {

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

        // Register service on port 2992

        ServerSocket s =

        new ServerSocket(2992);

        System.out.println("waiting .....");

        Socket s1=s.accept();

        //Wait and accept a connection

        //Get a communication stream        //associated

        //with the socket

        InputStream s1In = s1.getInputStream();

        DataInputStream dis = new DataInputStream(s1In);

        String st = new String (dis.readUTF());

        System.out.println(st);

        OutputStream s1out = s1.getOutputStream();

```

```

DataOutputStream dos =
    new DataOutputStream (s1.getOutputStream());

// Send a string!
dos.writeUTF(st.toUpperCase());


// Close the connection,
    //but not the server socket
dos.close();
s1out.close();
s1.close();
}
}

```

7. Write a client-server UDP socket program in java that the server listens for connection requests, and whatever message within 1024 bytes the client sends, the server converts it to uppercase and sends it back after 6 ms.

Client:

```

import java.io.*;
import java.net.*;

public class UDPclient
{
    public static void main(String args[]) throws Exception
    {
        System.out.println("write something ");

        BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));

        DatagramSocket clientSocket = new DatagramSocket();

        InetAddress IPAddress = InetAddress.getByName("localhost");
    }
}

```

```
byte[] sendData = new byte[1024];
byte[] receiveData = new byte[1024];
String sentence = inFromUser.readLine();
```

```
sendData = sentence.getBytes();
```

```
DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress,
9876);
```

```
clientSocket.send(sendPacket);
```

```
DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);
```

```
System.out.println("Waiting..... " );
```

```
clientSocket.receive(receivePacket);
```

```
System.out.println("Data Received..... " );
```

```
String modifiedSentence = new String(receivePacket.getData());
```

```
System.out.println("FROM SERVER:" + modifiedSentence);
```

```
clientSocket.close();
```

```
}
```

```
}
```

Server:

```
import java.io.*;
```

```
import java.net.*;
```

```
public class UDPserver
```

```
{
```

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

```
    {
```

```
        DatagramSocket serverSocket = new DatagramSocket(9876);
```

```

byte[] receiveData = new byte[1024];
byte[] sendData = new byte[1024];
while(true)
{
    DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);

    System.out.println("Waiting..... " );
    serverSocket.receive(receivePacket);

    System.out.println("Data Received..... " );
    String sentence = new String( receivePacket.getData());
    System.out.println("RECEIVED: " + sentence);
    Thread.sleep(3000)
;        InetAddress IPAddress = receivePacket.getAddress();
    int port = receivePacket.getPort();
    // int fa=receivePacket.getLength();

    String capitalizedSentence = sentence.toUpperCase();
    // int Sentencesize = sentence.length();

    sendData = capitalizedSentence.getBytes();

    DatagramPacket sendPacket = new DatagramPacket(sendData,sendData.length
,IPAddress, port);

    serverSocket.send(sendPacket);
}
}
}

```


8. Suppose we have an MS Access database named ICE_PUST which has a table named by Student with field (Name, Email, and Phone). Using this database answer the following questions:

a. Write down a java program to insert data into the Student table of the ICE_PUST database.

```
import java.sql.*;
public class java_sql_db_insert_data {
    public static void main(String[] args){
        try{
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");//Loading Driver
            Connection connection=
            DriverManager.getConnection("jdbc:ucanaccess://E:\\Database\\ICE_PUST.accdb");//Estab
            lishing Connection
            System.out.println("Connected Successfully");
            //Crating PreparedStatement object
            PreparedStatement preparedStatement=connection.prepareStatement("
            + "INSERT INTO Student(Name,Email,Phone) VALUES (?, ?,
            ?)");

            //Setting values for Each Parameter
            preparedStatement.setString(1, "Suhannn");
            preparedStatement.setString(2, "rabbi.ru@gmail.com");
            preparedStatement.setString(3, "01722633140");
            //Executing Query
            preparedStatement.executeUpdate();
            System.out.println("data inserted successfully");

        }catch(Exception e){
            System.out.println("Error in connection");
        }
    }
}
```

b. Create a java program to print all student records from the Student table of the ICE_PUST database.

```
import java.sql.*;
public class java_sql_db_show_data {
    public static void main(String[] args){
        try{
            Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");//Loading Driver
            Connection connection=
            DriverManager.getConnection("jdbc:ucanaccess://E:\\Database\\ICE_PUST.accdb");//Estab
            lishing Connection
            System.out.println("Connected Successfully");
            //Using SQL SELECT Query
            PreparedStatement preparedStatement=connection.prepareStatement("select
            * from Student");
```

```

        //Creating Java ResultSet object
        ResultSet resultSet=preparedStatement.executeQuery();
        System.out.println(resultSet);
        System.out.println("Name      "+"Email  "+"Phone");
        while(resultSet.next()){
            String name=resultSet.getString("Name");
            String email=resultSet.getString("Email");
            String phone=resultSet.getString("Phone");
            //Printing Results
            System.out.println(name+" "+email+" "+phone);
        }

    }catch(Exception e){
        System.out.println("Error in connection");
    }

}
}
}

```

9. Consider an MS access database named Lab_18 which has a table named by Teacher with field (Name, Email, and Phone). Using this database give the answer of the following questions:

- a. Write down a java program to create a GUI registration form according to the Teacher table.**
- b. Create a java program to insert data into the Teacher table of the Lab_18 database from the GUI registration form which you have already created.**

```

package suhan38;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
import javax.swing.*;
public class java_sql_gui_1_simple extends JFrame implements ActionListener
{
    JTextField tf1,tf2,tf3;
    JLabel l1,l2,l3;
    JButton b2;
    Connection con;
    ResultSet rs;
    Statement st;
    public java_sql_gui_1_simple()
    {
        super("persionDetails");
        setLayout(null);
    }
}

```

```

Label d=new Label("Person Details");
d.setBounds(200,10,150,50);
d.setBackground(Color.RED);
add(d);
l1=new JLabel("Name");
l1.setBounds(100,100,100,50);
add(l1);
tf1=new JTextField(" ");

tf1.setBounds(200,100,250,40);
add(tf1);
l2=new JLabel("Email");
l2.setBounds(100,150,100,50);
add(l2);
tf2=new JTextField();
tf2.setBounds(200,150,250,40);
add(tf2);
l3=new JLabel("Phone");
l3.setBounds(100,200,100,50);

//l3.setFont(new Font("",Font.BOLD));
add(l3);
tf3=new JTextField();
tf3.setBounds(200,200,250,40);
add(tf3);
b2=new JButton("Insert");
b2.setBounds(100,250,90,50);
add(b2);

b2.addActionListener(this);

setSize(600,600);
setVisible(true);
connection();
}

public void connection()
{
    try
    {
        Class.forName("net.ucanaccess.jdbc.UcanaccessDriver");//Loading Driver
        con=
DriverManager.getConnection("jdbc:ucanaccess://E:\\Database\\Lab_18.accdb");//Establishing Connection
        System.out.println("Connected Successfully");
        st=con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_UPD
TABLE);
        rs=st.executeQuery("SELECT * FROM Teacher");
        System.out.println("connect");

        //JOptionPane.showMessageDialog(null, "please click next Button");
    }catch(Exception e)
    {
        System.out.println(e);
    }
}

```

```
}
```

```
public void actionPerformed(ActionEvent e)
{

    if(e.getSource()==b2)
    {
        System.out.println("I am Insert");
        String up1=tf1.getText();
        String up2=tf2.getText();
        String up3=tf3.getText();
        try
        {
            rs.updateString("Name",up1);
            rs.updateString("Email",up2);
            rs.updateString("Phone",up3);
            rs.insertRow();
            System.out.println("one row is inserted");

        }
        catch(Exception er)
        {
        }
    }

}
```

```
public static void main(String a[])
{
    new java_sql_gui_1_simple();
}
}
```