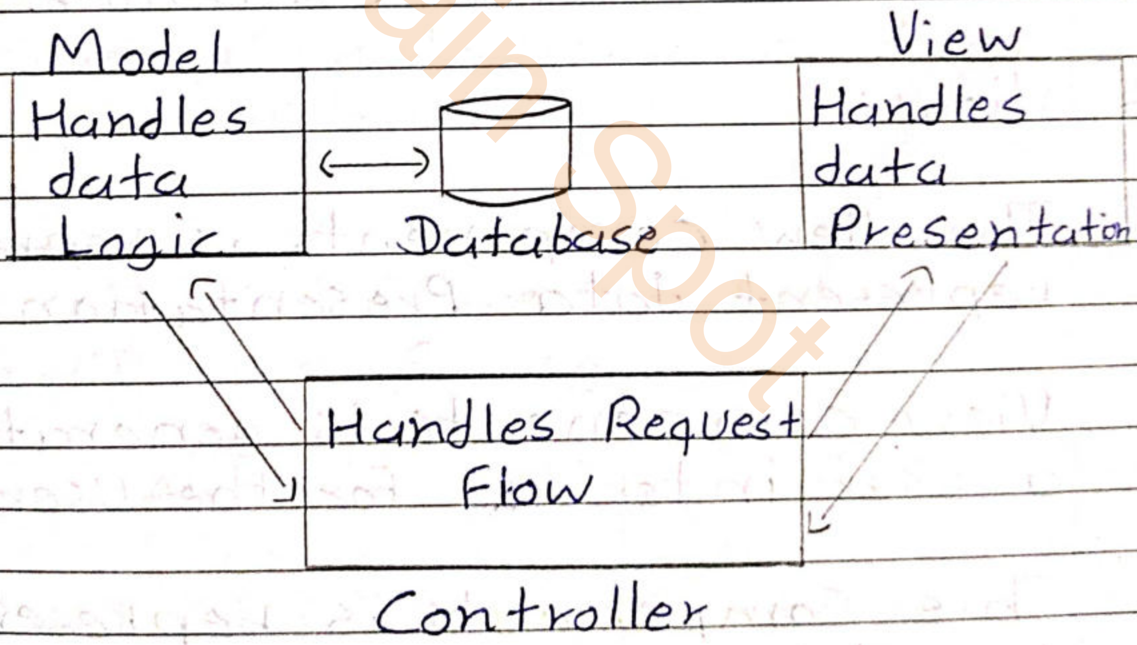
1 Explain MVC Architecture.

=> MVC stands for Model View Controller which is represent design pattern that sparates application.

MVC Framework divided Application into the three part:

(a) Model
(b) View
(c) Controller



a Model:

The Model component contains all the data-related logic

The model components can access the all the data releted Logic in database.

It can Represent how data transfer between view and Controller or between any data Logic.

Model Components is responds to the controller's request and also response to the Controller.

Model components can transfer the data to the Controller.

b View :

The View components is ~~represet~~ represent data Presentation

View components is generates a User interface for the User.

This Components is represent how data will look to the User.

View Components is responds to the controller's request for data presentation.

This View components is only interacts with the controller.

c Controller:

The controller is the enable interconnection between Model and View.

Controller can ~~be~~ do Request to the model component and view component.

The Controller does not have any type of data logic or data representation.

2 Explain Difference between AWT and Swing.

	AWT	Swing
1	AWT Stands For Abstract Windows toolkit.	Swing is called as Java Foundation classes.
2	AWT does not Support MVC.	Swing is Support MVC.

Date : / /

	AWT	Swing
3	AWT has less powerful Compone-nts.	Swing has more Powerful Components
4	More Execution Time.	Less Execution Time.
5	Heavy Weighted	Light Weighted
6	Platform dependent Components.	Platform independent components.
7	Require java. awt package.	Require javax. swing package

3 Write different Swing events and explain with example.

=> The change in the state of an Object behavior by performing actions is called event.

For Performing event in Swing we have to add ~~im~~ java. awt. event packege.

There are different type of event in Swing.

These are the Basic Swing events.

a) Key event
b) Mouse event
c) Action event
d) Item event
e) Focus event
f) Text event

a Key Event:

Key event is used to perform action using Keyboard.

For Performing the Key event, We have to use KeyListener.

There are different types of Keyboard event.

Ex. KeyPressed (KeyEvent e)
KeyReleased (KeyEvent e)
KeyTyped (KeyEvent e)

b Mouse Event:

Mouse event is used to perform

action using Mouse.

For Performing the Mouse event, we have to use MouseListener.

~~Ther~~
There are different types of Mouse event in swing.

Ex. mouseClicked(MouseEvent e)
mouseEntered(MouseEvent e)
mouseExited(MouseEvent e)
mousePressed(MouseEvent e)

c Action event:

When we click any button or menu item than action event is Performed.

For Performing the Action event, we have to use ActionListener.

There is Only One Method to Perform Action event.

Ex. actionPerformed(ActionEvent e)

d Item Event :

When we click on the check box, than item event is perform.

For Performing the Item event, we have to use ItemListener.

There is only one method to Perform Item event.

itemStateChanged(ItemEvent e)

e Focus Event :

When Component gain or losses the Keyboard or Mouse focus than Focus event is performed.

For Performing the Focus event, we have to use FocusListener.

There are different types of Focus event in swing.

Ex. FocusGained(FocusEvent e)
FocusLost(FocusEvent e)

F Text Event:

When an Object's text is changed than text event is performed.

For Performing text event, we have to Use TextListener.

There is only one method for Performed text event.

textChanged(TextEvent e).

Ex.

```
import java.awt.*;
import java.awt.event;

class java extends Frame
 implements ActionListener
{
  TextField tf = new TextField();

  java()
  {
    tf.setBounds(60,50,170,20);
    add(tf);
    setSize(300,300);
  }
public void actionPerformed
        (ActionEvent e)
{
    tf.setText("khushi");
}
public static Void main(String, args[])
{
    new java();
 }

}
```

4 Explain Swing GUI Components :

(a) JComponent :

The JComponent class is the base class of all swing components except top-events containers.

Only JFrame and JDialog can not inherit JComponent class.

JComponent class extends the Container class which extends Component.

Syntax For Create JComponent class :

```
class class extends JComponent
      name
{
   //code
}
```

(b) JButton :

JButton is a labeled Component which is use to performed click event.

Using JButton we can performed Mouse event and Key event.

Syntax For Create Button :

```
JButton JButton = new JButton
        Object      ("Button
                      name");
```

Using this method, we can create JButton with its name.

(c) JLabel :

JLabel is a Component which is use to set text in the container.

JLabel will displays text in the form of Read-only text in container

Syntax For Create JLabel :

```
JLabel JLabel = new JLabel("Text");
       Object
```

:
Using this syntax, we can set any type of text in Container.

(F) JTextComponent :

JTextComponent is an abstract class that serves as the base class for all text-based swing component.

Using JTextComponent abstract class we can add JTextField, JTextArea and JEditorPane in a Container.

(K) JCheckBox :

JCheckBox is Used to create a checkBox in the container.

This Box can get a value either

true or False by checking and unchecking the checkbox.

Checkbox allows multiple option For check or uncheck in the checkbox.

Syntax For Create CheckBox :

JCheckBox JCheckBox = new Object
JCheckBox("Value", true or False);

(J) JRadioButton :

JRadioButton is use to select only one option from the multiple option in container.

JRadioButton allows Only one option for check in the container.

It should be added in Button Group to select one radio button only.

Page No.
Date : / /

Syntax For Create JRadioButton :

JRadioButton JRadioButton = new
Object

JRadioButton ("Value");

(O) J ToggleButton :

J Toggle Button is used to create toggle button which is two-state button.

Toggle Button have only two Option either on or off.

Syntax For Create JToggle Button :

JToggleButton JToggleButton =
Object

new JToggleButton ("ON or OFF")

5 Explain various layout managers with example.

=> The Layout Managers are used to arrange components in a particular manner.

The Java Layout Manager control the positioning and size of the components in GUI forms.

There are three types of Layout Managers.

1) Border Layout
2) Flow Layout
3) Grid Layout

1 Border Layout :

The Border Layout is used to arrange the components in Five regions : North, South, East, Wast and Center.

Each region may contain one component only.

We can create Border Layout with two type.

(i) BorderLayout ( ) : Create border Layout with no gaps between the components.

(ii) BorderLayout(int hgap, int vgap): Create Border Layout with horizontal and vertical gaps between the components.

2 Flow Layout :

The Java Flow Layout is used to arrange the components in a line means in one flow.

We can create three types Flow Layout:

(i) FlowLayout ( ) : Flow Layout with centered alignment.

(ii) FlowLayout (int align) : Flow Layout with given alignment.

(iii) FlowLayout (int align, int hgap, int vgap) : Flow Layout with given horizontal alignment and given vertical gap.

SMVS
Page No.
Date : / /

3 Grid Layout ::

The Java Grid Layout is used to arrange the components in a rectangular grid.

One component is display in each rectangle.

We can Create Grid Layout in Three ways.

(i) GridLayout() : Create a grid Layout with one column per component in a row.

(ii) GridLayout(int rows, int ~~clo~~ columns
Create a grid Layout with the given row and columns.

(iii) GridLayout(int rows, int columns, int hgap, int vgap) : Create a grid Layout with given row and column and given horizontal and vertical gaps.

Ex.

```
import java.awt.*;
import java.swing.*;

class Java extends JFrame
{

    JPanel  P1 = new JPanel();
    JPanel  P2 = new JPanel();
    JButton b1 = new JButton();
    JButton b2  = new JButton();
    JLabel  l1 = new JLabel("khushi")
    JLabel  l2 = new JLabel("khush")

    P1.setLayout(new GridLayout(4,4))
    b1. add (b1);
    b2. add (b2);

    P2. setLayout(new BorderLayout())
    P2. add (l1, east);
    P2. add (l2, west);

    Content Pane = getContentPane()
     .setLayout(new BorderLayout());

     Pane. add ( P1, north);
     Pane. add (P2. south);

    public static void main (String,
     args[])
```

```
        {
            new Java();
        }

}
```

6 Explain window and frame with respect to Swing.

=) Frame:

Frame is an AWT Component which is a top level window.

Frame have a title bar and Broder.

Frame content all tha awt component like button, label or Panel.

Frame can content multiple Panel in one Frame but Frame can content only one Frame.

We can not add multiple Frame in One Frame.

Example :

```
import java.awt.*;
import java.swing.*;

class Java
{
  Java()
  {
   JFrame F = new JFrame();
   JPanel P = new JPanel();

     P.setBounds(40,80,200,200);
     P.setBackground(color.red);
     F.add(P);
     F.setsize(400,400);
  }

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

}
```

→ Window :

Window is a part of Java swing

and it can appear on any part of the user desktop.

The window is a container that does not include borders and menu bar.

We can add Multiple Window in one Frame.

Window can containes button or label in One Frame.

Example:

```
import java.awt.*;
import java.swing.*;

class Java
{  Java()
 { JFrame F = new JFrame();
   JWindow W = new JWindow();
   JLabel l = new JLabel("khushi");

     W.add(l);
     W.setsize(200, 100);
     F.add(W);
     F.setsize(400, 400);
 }
public static void main(String,
      args[])
  {
      new Java();
   }
}
```