8a) Problem statement :- write a java program that import and use the user defined packages.

Source code :-

```
Greeting mypackage;
public class Greetings
{ public void sayHello()
    { System.out.println ("Hello from the Greetings
      class in my package!");
} }
```

Main program:-

```
import mypackage.Greeting;
public class Mainprogram
{ public Static void main (String args[])
    { Greeting greeting =new Greeting();
      greeting.SayHello();
} }
```

Compile package :

javac mypackage\Greeting.java

Compile the main program

Javac MainProgram.java

Run the program

Java Mainprogram

Output :- ~~Javamain program~~

Hello Java the greeting class in mypackage!

5) b Problem statement :- without writing any code, build a GUI that display text in label and image in an Imagview (use JavaFx)

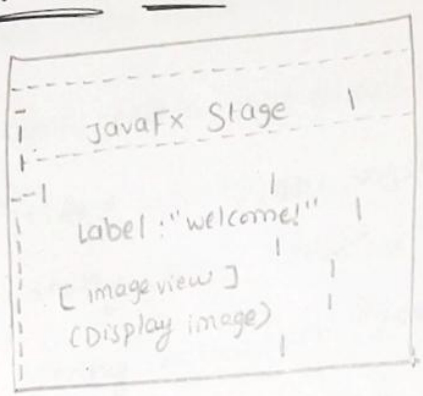
Step by step Design outline for JavaFx GUI

1. Create a Main window (Stage):
   - The main window (stage) will contain a layout pane to organize the UI components.
2. use a layout pane (VBOX):
   - A VBox (vertical box) is a suitable layout pane to stack components vertically.
   - Set spacing and alignment properties on the VBOX to control the arrangement of the label and image
3. Add a label :-
   - place a label at the top of the VBOX to display text.
   - Set the text content of the label, such as "welcome to JavaFx!" or any other text.
4. Add an Imageview:
   - Below the label, add an image view to display an image
   - Set the imageview properties such as fitwidth, fitHeight, or preserveRato to control how the image appears.
   - use an image object (loaded from a URL or file path) to set the imageview

5. Setup the Scene and Display :-

- Add the VBox (with the label and image view) to the scene
- Display the stage with the scene.

Visual layout :-



6c) Problem Statement:- Build a Tip Calculator app using several JavaFX components and learn how to respond to user interactions with GUI

Step by Step Design and Functionality outline:

1. Setup the main stage and scene:
   Use a VBox layout to stack elements vertically setting spacing and alignment for a clean look.
2. Create the UI components;
   - Labels: provide text prompts for each field like "Enter Bill Amount" and select Tip percentage".
   - TextField: Allow users to input the bill amount
   - ComboBox: offer a dropdown with different tip percentages (eg, 10%, 15%, 20%)
   - Button; A "calculate" button to compute the total
   - Result label: Display the calculated tip amount and total cost.
3. Add event Handling:-
   - Use event handling for the "calculate" button to perform the calculation based on user input and selection.
   - parse input from the textfield and combobox calculate the tip and total & display the results in the result label.
4. Error Handling:
   Handle invalid inputs (eg., non-neumaric input for the bill amount) to avoid application crashes.

Code Structure:-

1. Layout & Design
   - Stage: The main window where the components will be displayed.
   - VBox: Used to stack Components with spacing
   - HBoxes:- Nested within the VBox for organizing Components horizontally. eg:- label + Textfield in one row.
2. Component List
   - Label: "Enter Bill Amount", "Select Tip percentage", "Tip Amount", "Total Amount".
   - TextField: For entering the bill amount
   - ComboBox: For selecting the tip percentage
   - Button: "Calculate" button to perform the calculation
   - Result Label: to display the tip amount and total amount
3. Event Handling logic:
   - Retrive the entered bill amount from the Text field.
   - Retrieve the selected tip percentage from the Combobox.
   - calculate the tip by multiplying the bill amount with the selected percentage.
   - calculate the total by adding the tip to the bill amount.
   - Display the calculated tip and total in the Result label.

Visual layout of the tip Calculator App.

```
-----------------------------------
---------
|        Tip Calculator
|
|---------------------------------
-----------|
| Enter Bill Amount :
[ TextField              ]|
| Select Tip percentage :
[ ComboBox : 10%, 15%, 20% ]|
|                                |
|              [Calculate Button]
|
| Tip amount :        $ calculated
TIP]   |
| Total amount :      $ calculated
Total]  |
-----------------------------------
----------
```

9a) Problem statement:- write a java program that connects to data base using JDBC

Source code:-

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
public class DatabaseConnectionExample
{ public static void main (String args[])
  { String url = "jdbc:mysql://localhost:3306/testdb";
    String user = "root";
    String password = "your password";
    Connection connection = null;
    try {
        Connection = DriverManager.getConnection
                        (url, user, password);
        System.out.println ("Database Connection
                        successfully.");
    }
    catch (SQLException)
    { System.out.println ("failed to connect to
      the database");
      e.printStackTrace();
    }
    finally
    { try { if (connection != null)
            {
              Connection.close();
              System.out.println("Database
                        Connection closed");
            } }
```

```
    catch (SQLException ex)
    {
        ex.printStrackTrace();
    }
}
}
```

output :-

Successful Connection :-

Database Connection successful.

Database Connection Closed

failed connection :-

Failed to connect to the database

Com.mysql.cj.jdbc.exceptions.CommunicationsException:

Communications link failure

The last packet sent successfully to the server was 0 milliseconds ago. The driver has not recieved any packets from the server at.....

9b) Problem Statement :- write a java program to connect to a data base using JDBC and insert values into it.

Source code :

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.Exception;
public class InsertDataExample
{ public static void main(String args[])
  { String url = "jdbc:mysql://localhost:3306/testdb";
    String user = "root";
    String password = "yourpassword";
    String sqlInsert = "INSERT INTO employees (id,
                        name, position) values (?,?,?)";
    Connection connection = null;
    PreparedStatement preparedStatement = null;
  try {
      connection = DriverManager.getConnection(url, user,
                                   password);
    System.out.println("Connected to the database");
    preparedStatement = connection.PreparedStatement
                                   (sqlInsert)
    preparedstatement.setInt(1,1);
    preparedStatement.setString(2, "JohnDoe");
    preparedStatement.setString(3, "Software Engineering");
    int rowInserted = preparedStatement.executeUpdate();
    if (rowInserted > 0)
    {
```

```
    {
        System.out.println("A new employee was inserted
                                successfully");
    }
}
Catch ( SQLException e)
{   System.out.println("Data base error occured");
    e.printStackTrace();
}
finally
{   try{ if (preparedStatement != null) preparedStatement.
                                        Closed();
         if (Connection != NULL) Connection.Close();
             System.out.println(" Database Connection
                                        Closed");
         }
         catch (SQLException ex)
         {   ex.printStackTrace();
         }
}}}
```

Output :-

Connected to the database
A new employee was inserted successfully
Database Connection Closed

9c) problem statement:- write a java program to connect to a database using JDBC and delete values from it.

source code:-

```
import java.sql.Connection;
import java.sq.DriverManager;
import java.sql.PrepareStatement;
import java.sql.SQLException;
public class DatabaseDeleteExample
{
    private static final String DB_URL =
    "jdbc:mysql://localhost:3306/your-database-name";
    private static final String USERNAME = "your-username"
    private static final string PASSWORD = "yourpassword";
    public static void main(String[] args)
    {
        Connection connection = null;
        PreparedStatement preparedstatement = null;
        try {
            connection = DriverManager.getConnection(DB_URL,
                                USERNAME, PASSWORD);
        System.out.println("Connected to the database
                                successfully.");
        String sql = "DELETE FROM your_table_name
                    WHERE Column_name = ?";
        PreparedStatement = Connection.preparedStatement(sql);
        PreparedStatement.SetString(1, "value_to_delete");
        int rowsAffected = PrepareStatement.executeUpdate();
        System.out.println(rowsAffected + " rows) deleted.");
    }
```

```
catch (SQLException e)
{
    e.printStackTrace();
}
Finally {
    try {
        if (preparedStatement != null)
            preparedStatement.close();
        if (connection != null)
            connection.close();
    }
    catch (SQLException e)
    {
        e.printStackTrace();
    }
}
}
}
```

output:-

connected to the database successfully

1 row(s) deleted