

Assignment no:- 1

Installation and setup of java development kit(JDK),setup android SDK,setup android studio,setup android development tools ADT plugins create android virtual device

1. Install Java Development Kit (JDK)

Step 1: Go to the [Oracle JDK download page](#).

Step 2: Download the latest version of JDK for your operating system (Windows, macOS, or Linux).

Step 3: Run the downloaded installer and follow the on-screen instructions to install JDK.

Step 4: After installation, open a terminal (or command prompt) and verify the installation by typing:

```
java -version
javac -version
```

This should display the installed version of Java.

Step 5: Set up the JAVA_HOME environment variable:

- **Windows:**
 1. Right-click on "My Computer" → Properties → Advanced system settings → Environment Variables.
 2. Under **System Variables**, click **New**. Name the variable JAVA_HOME and set its value to the directory where JDK is installed (e.g., C:\Program Files\Java\jdk-<version>).
 3. Add %JAVA_HOME%\bin to the Path system variable.
- **macOS/Linux:**
 - o Open a terminal and add the following line to your .bash_profile, .bashrc, OR .zshrc:

```
bash
export JAVA_HOME=/path/to/jdk
export PATH=$JAVA_HOME/bin:$PATH
```

2. Install Android SDK

Step 1: Download Android Studio, which includes the Android SDK, from the [official Android Studio website](#).

Step 2: Run the installer and follow the on-screen instructions to install Android Studio. During the installation, make sure that the Android SDK is included in the setup process.

Step 3: Once installed, open Android Studio.

Step 4: On the first launch, Android Studio will guide you through downloading necessary SDK components.

Step 5: To update or install additional SDK components, go to:

- **Tools → SDK Manager →** Install or update required SDK packages.

Step 6: Set up the ANDROID_HOME environment variable:

- **Windows:**
 1. Right-click "My Computer" → Properties → Advanced system settings → Environment Variables.
 2. Add a new **System Variable** named ANDROID_HOME, and set its value to the path where the Android SDK is installed (e.g., C:\Users\YourUserName\AppData\Local\Android\Sdk).
 3. Add %ANDROID_HOME%\tools and %ANDROID_HOME%\platform-tools to the Path system variable.
- **macOS/Linux:**
 - o Add the following lines to your .bash_profile, .bashrc, OR .zshrc file:

```
bash
export ANDROID_HOME=/path/to/android/sdk
export PATH=$ANDROID_HOME/tools:$ANDROID_HOME/platform-tools:$PATH
```

3. Install and Set Up Android Studio

Step 1: Download Android Studio from the [official website](#).

Step 2: Run the installer for your operating system.

Step 3: After installation, launch Android Studio.

Step 4: On the first launch, Android Studio will ask you to install the necessary SDK components. It will also download essential Android development tools (like the SDK Manager).

Step 5: Choose the "Standard" installation type during setup.

Step 6: Once Android Studio opens, configure the default SDK location (if not already configured) by navigating to **File → Settings** (on Windows) or **Android Studio → Preferences** (on macOS) → **Appearance & Behavior → System Settings → Android SDK**.

Step 7: Make sure the SDK components are up to date by checking for updates under **Tools → SDK Manager**.

4. Install Android Development Tools (ADT) Plugin (For Eclipse Setup Only)

Note: ADT Plugin is no longer required for Android Studio. Android Studio provides built-in Android development tools. However, if you are using Eclipse for some reason, you can follow these steps.

Step 1: Open Eclipse IDE and go to **Help** → **Eclipse Marketplace**.

Step 2: In the Marketplace, search for "ADT" and install the **Android Development Tools (ADT)** plugin.

Step 3: Follow the prompts to install the plugin. After installation, restart Eclipse.

Step 4: In Eclipse, go to **Window** → **Preferences**, and under **Android**, set the SDK location to the Android SDK folder.

5. Create an Android Virtual Device (AVD)

Step 1: Open Android Studio.

Step 2: In Android Studio, go to **Tools** → **AVD Manager**.

Step 3: Click on **Create Virtual Device**.

Step 4: Choose a device type (e.g., Pixel, Nexus 5) and click **Next**.

Step 5: Select a system image for the virtual device. If needed, download a system image (e.g., for Android 11, etc.).

Step 6: Choose a system image and click **Next**.

Step 7: Configure the AVD settings such as device RAM, scale, and orientation.

Step 8: Click **Finish** to create the AVD.

Step 9: Once the AVD is created, it will appear in the AVD Manager. Click **Start** to run your virtual device.

Assignment no:- 2

Create “Hello World” application. That will display “Hello World” in the middle of the screen using TextView Widget in the red color.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:gravity="center"
    android:background="#FFFFFF">

    <!-- TextView displaying Hello World -->
    <TextView
        android:id="@+id/hello_world_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:textColor="#FF0000" <!-- Red color -->
        android:textSize="30sp" />

</LinearLayout>
```

MainActivity.java

```
package com.example.helloworldapp;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
import android.widget.TextView;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Set the layout to activity_main.xml

        // Optional: You can programmatically access and modify the TextView if needed
        TextView helloWorldText = findViewById(R.id.hello_world_text);
        helloWorldText.setText("Hello World");
    }
}
```

Assignment no:- 3

Create registration page to demonstrate of Basic widgets available in android

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp">

    <!-- Title -->
    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Registration Form"
        android:textSize="24sp"
        android:layout_gravity="center"
        android:paddingBottom="20dp" />

    <!-- Name EditText -->
    <TextView
        android:id="@+id/name_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Full Name:"
        android:textSize="16sp" />

    <EditText
        android:id="@+id/name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your full name"
        android:inputType="textPersonName"
        android:layout_marginBottom="16dp" />

    <!-- Email EditText -->
    <TextView
        android:id="@+id/email_label"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Email:"
        android:textSize="16sp" />

    <EditText
        android:id="@+id/email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your email"
        android:inputType="textEmailAddress"
        android:layout_marginBottom="16dp" />
```

```

<!-- Gender RadioButtons -->
<TextView
    android:id="@+id/gender_label"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Gender:"
    android:textSize="16sp" />

<RadioGroup
    android:id="@+id/gender_group"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:layout_marginBottom="16dp">

    <RadioButton
        android:id="@+id/male"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Male" />

    <RadioButton
        android:id="@+id/female"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Female" />
</RadioGroup>

<!-- Terms CheckBox -->
<CheckBox
    android:id="@+id/terms_checkbox"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="I accept the terms and conditions"
    android:layout_marginBottom="16dp" />

<!-- Register Button -->
<Button
    android:id="@+id/register_button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Register"
    android:textSize="18sp" />

</LinearLayout>

```

MainActivity.java

```
package com.example.registrationpage;
```

```

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.RadioButton;

```

```

import android.widget.RadioGroup;
import android.widget.Toast;
import android.widget.CheckBox;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    EditText nameEditText, emailEditText;
    RadioGroup genderGroup;
    RadioButton maleRadioButton, femaleRadioButton;
    CheckBox termsCheckbox;
    Button registerButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize the views
        nameEditText = findViewById(R.id.name);
        emailEditText = findViewById(R.id.email);
        genderGroup = findViewById(R.id.gender_group);
        maleRadioButton = findViewById(R.id.male);
        femaleRadioButton = findViewById(R.id.female);
        termsCheckbox = findViewById(R.id.terms_checkbox);
        registerButton = findViewById(R.id.register_button);

        // Set an OnClickListener for the Register button
        registerButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Validate the inputs
                String name = nameEditText.getText().toString();
                String email = emailEditText.getText().toString();

                if (name.isEmpty() || email.isEmpty()) {
                    Toast.makeText(MainActivity.this, "Please fill in all fields", Toast.LENGTH_SHORT).show();
                    return;
                }

                if (!termsCheckbox.isChecked()) {
                    Toast.makeText(MainActivity.this, "You must accept the terms and conditions",
Toast.LENGTH_SHORT).show();
                    return;
                }

                int selectedGenderId = genderGroup.getCheckedRadioButtonId();
                if (selectedGenderId == -1) {
                    Toast.makeText(MainActivity.this, "Please select a gender", Toast.LENGTH_SHORT).show();
                    return;
                }

                String gender = (selectedGenderId == R.id.male) ? "Male" : "Female";

```

```
        // If everything is valid, show a Toast message
        String message = "Name: " + name + "\nEmail: " + email + "\nGender: " + gender;
        Toast.makeText(MainActivity.this, message, Toast.LENGTH_LONG).show();
    }
});
}
```

Assignment no:- 4

Create sample application with login module (check username and password on successful login, change TextView "Login Successful". and on failing login alert user using Toast "Login Fail").

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="32dp"
    android:gravity="center">

    <!-- Title -->
    <TextView
        android:id="@+id/title"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:textSize="24sp"
        android:layout_marginBottom="20dp"
        android:gravity="center"/>

    <!-- Username EditText -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Username:"
        android:textSize="16sp"/>

    <EditText
        android:id="@+id/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your username"
        android:inputType="textPersonName"
        android:layout_marginBottom="16dp"/>

    <!-- Password EditText -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password:"
        android:textSize="16sp"/>

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your password"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>

    <!-- Login Button -->
    <Button
```



```

        android:id="@+id/login_button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login"
        android:textSize="18sp"/>

```

```

<!-- Status TextView -->

```

```

<TextView
    android:id="@+id/status"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="18sp"
    android:textColor="#32CD32" <!-- Default green color -->
    android:layout_marginTop="20dp"/>

```

```

</LinearLayout>

```

MainActivity.java (Logic for Validation)

```

package com.example.loginapp;

```

```

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

```

```

public class MainActivity extends AppCompatActivity {

```

```

    // Declare UI elements
    EditText usernameEditText, passwordEditText;
    Button loginButton;
    TextView statusTextView;

```

```

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Set the layout to activity_main.xml

```

```

        // Initialize UI elements
        usernameEditText = findViewById(R.id.username);
        passwordEditText = findViewById(R.id.password);
        loginButton = findViewById(R.id.login_button);
        statusTextView = findViewById(R.id.status);

```

```

        // Hardcoded credentials for demonstration
        final String correctUsername = "user123";
        final String correctPassword = "pass123";

```

```

        // Set an OnClickListener for the login button
        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

                public void onClick(View v) {
                    // Get the entered username and password
                    String enteredUsername = usernameEditText.getText().toString();
                    String enteredPassword = passwordEditText.getText().toString();

```

```

                    // Check if the entered credentials match the correct ones
                    if (enteredUsername.equals(correctUsername) && enteredPassword.equals(correctPassword)) {
                        // If login is successful, change TextView text and display success message
                        statusTextView.setText("Login Successful");
                        statusTextView.setTextColor(getResources().getColor(android.R.color.holo_green_dark)); // Set text color to green

```

```

                    } else {
                        // If login fails, show a Toast message

```

```
        Toast.makeText(MainActivity.this, "Login Fail", Toast.LENGTH_SHORT).show();
    }
}    }); }}
```

Assignment No:- 5

Create an application for demonstration of 'Scroll view' in android?

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingBottom="32dp">

        <!-- Title Text -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="ScrollView Demonstration"
            android:textSize="24sp"
            android:layout_gravity="center"
            android:layout_marginBottom="20dp" />

        <!-- TextView for content -->
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="This is a simple demonstration of ScrollView. " +
                "Scroll down to see more content. " +
                "ScrollView allows you to add content that overflows the screen and makes it scrollable."
            android:textSize="18sp"
            android:layout_marginBottom="20dp" />

        <!-- More content to demonstrate scrolling -->
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="This is some additional content to make the screen longer. " +
                "You can keep adding more content to test scrolling."
            android:textSize="18sp"
            android:layout_marginBottom="20dp" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="Keep scrolling to see more examples of how you can fit long content " +
                "into a limited screen space using ScrollView."
            android:textSize="18sp"
            android:layout_marginBottom="20dp" />

    </LinearLayout>

</ScrollView>
```

```

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="This is more text. You can add as much content as needed to test scrolling
behavior."
    android:textSize="18sp"
    android:layout_marginBottom="20dp" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="ScrollViews are often used for displaying long lists, forms, or content that exceeds
the screen height."
    android:textSize="18sp"
    android:layout_marginBottom="20dp" />

</LinearLayout>
</ScrollView>

```

MainActivity.java (Logic)

```

package com.example.scrollviewapp;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Set the layout to activity_main.xml
    }
}

```

Assignment no:-6

Create login application where you will have to validate username and passwords till the username and password is not validated , login button should remain disabled.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="32dp"
    android:gravity="center">

    <!-- Title Text -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login"
        android:textSize="24sp"
        android:layout_marginBottom="20dp"
        android:gravity="center"/>

    <!-- Username EditText -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Username:"
        android:textSize="16sp"/>

    <EditText
        android:id="@+id/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your username"
        android:inputType="textPersonName"
        android:layout_marginBottom="16dp"/>

    <!-- Password EditText -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password:"
        android:textSize="16sp"/>

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your password"
        android:inputType="textPassword"
        android:layout_marginBottom="16dp"/>
```

```

<!-- Login Button -->
<Button
    android:id="@+id/login_button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Login"
    android:textSize="18sp"
    android:enabled="false"/> <!-- Initially disabled -->

<!-- Status TextView (Optional) -->
<TextView
    android:id="@+id/status"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=""
    android:textSize="18sp"
    android:layout_marginTop="20dp"
    android:textColor="#32CD32" /> <!-- Green color for success message -->

</LinearLayout>

```

MainActivity.java

```

package com.example.loginvalidationapp;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    // Declare UI elements
    EditText usernameEditText, passwordEditText;
    Button loginButton;
    TextView statusTextView;

    // Hardcoded credentials for demonstration
    final String correctUsername = "user123";
    final String correctPassword = "pass123";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Set the layout to activity_main.xml

        // Initialize UI elements
        usernameEditText = findViewById(R.id.username);
        passwordEditText = findViewById(R.id.password);
        loginButton = findViewById(R.id.login_button);
    }
}

```

```

statusTextView = findViewById(R.id.status);

// Disable Login button initially
loginButton.setEnabled(false);

// Add TextChangeListeners for the EditText fields
usernameEditText.addTextChangedListener(new android.text.TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int start, int count, int after) {}

    @Override
    public void onTextChanged(CharSequence charSequence, int start, int before, int count) {
        checkFields(); // Check if both fields are filled
    }

    @Override
    public void afterTextChanged(android.text.Editable editable) {}
});

passwordEditText.addTextChangedListener(new android.text.TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence charSequence, int start, int count, int after) {}

    @Override
    public void onTextChanged(CharSequence charSequence, int start, int before, int count) {
        checkFields(); // Check if both fields are filled
    }

    @Override
    public void afterTextChanged(android.text.Editable editable) {}
});

// Set the OnClickListener for the Login button
loginButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Get the entered username and password
        String enteredUsername = usernameEditText.getText().toString();
        String enteredPassword = passwordEditText.getText().toString();

        // Check if the credentials match the hardcoded values
        if (enteredUsername.equals(correctUsername) && enteredPassword.equals(correctPassword)) {
            // If login is successful, display a success message
            statusTextView.setText("Login Successful");
            Toast.makeText(MainActivity.this, "Welcome!", Toast.LENGTH_SHORT).show();
        } else {
            // If login fails, display a failure message
            statusTextView.setText("");
            Toast.makeText(MainActivity.this, "Login Failed", Toast.LENGTH_SHORT).show();
        }
    }
});
}

```

```
// Helper method to check if both fields are filled
private void checkFields() {
    String username = usernameEditText.getText().toString();
    String password = passwordEditText.getText().toString();

    // Enable the Login button only if both fields are non-empty
    if (!username.isEmpty() && !password.isEmpty()) {
        loginButton.setEnabled(true);
    } else {
        loginButton.setEnabled(false);
    }
}
}
```


Assignment no:- 7

Create an application for calculator?

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="16dp"
    android:gravity="center">

    <!-- Display Result TextView -->
    <TextView
        android:id="@+id/resultText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text=""
        android:textSize="32sp"
        android:gravity="end"
        android:padding="16dp"
        android:background="#E0E0E0"
        android:layout_marginBottom="20dp"
        android:editable="false"/>

    <!-- Buttons for calculator -->
    <GridLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:columnCount="4"
        android:rowCount="5"
        android:layout_gravity="center">

        <!-- Number and operator buttons -->
        <Button android:id="@+id/button7" android:text="7" style="@style/CalculatorButton"/>
        <Button android:id="@+id/button8" android:text="8" style="@style/CalculatorButton"/>
        <Button android:id="@+id/button9" android:text="9" style="@style/CalculatorButton"/>
        <Button android:id="@+id/buttonDiv" android:text="/" style="@style/CalculatorButton"/>

        <Button android:id="@+id/button4" android:text="4" style="@style/CalculatorButton"/>
        <Button android:id="@+id/button5" android:text="5" style="@style/CalculatorButton"/>
        <Button android:id="@+id/button6" android:text="6" style="@style/CalculatorButton"/>
        <Button android:id="@+id/buttonMul" android:text="*" style="@style/CalculatorButton"/>

        <Button android:id="@+id/button1" android:text="1" style="@style/CalculatorButton"/>
        <Button android:id="@+id/button2" android:text="2" style="@style/CalculatorButton"/>
        <Button android:id="@+id/button3" android:text="3" style="@style/CalculatorButton"/>
        <Button android:id="@+id/buttonSub" android:text="-" style="@style/CalculatorButton"/>

        <Button android:id="@+id/button0" android:text="0" style="@style/CalculatorButton"/>
        <Button android:id="@+id/buttonClear" android:text="C" style="@style/CalculatorButton"/>
        <Button android:id="@+id/buttonEqual" android:text="=" style="@style/CalculatorButton"/>
        <Button android:id="@+id/buttonAdd" android:text="+" style="@style/CalculatorButton"/>
    </GridLayout>
</LinearLayout>
```

```
</GridLayout>
</LinearLayout>
```

o define the **style for the buttons**, you can create a custom style inside res/values/styles.xml:

styles.xml:

```
<resources>
    <style name="CalculatorButton">
        <item name="android:layout_width">0dp</item>
        <item name="android:layout_height">wrap_content</item>
        <item name="android:layout_rowWeight">1</item>
        <item name="android:layout_columnWeight">1</item>
        <item name="android:padding">16dp</item>
        <item name="android:textSize">20sp</item> </style></resources>
```

MainActivity.java (Handling Logic)

```
package com.example.calculatorapp;
```

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.GridLayout;
import androidx.appcompat.app.AppCompatActivity;
```

```
public class MainActivity extends AppCompatActivity {
```

```
    private TextView resultText;
    private String currentInput = "";
    private String operator = "";
    private double firstOperand = 0;
    private boolean isOperatorClicked = false;
```

```
@Override
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
```

```
    // Initialize UI elements
    resultText = findViewById(R.id.resultText);
```

```
    // Number buttons
```

```
    setNumberButton(R.id.button0, "0");
    setNumberButton(R.id.button1, "1");
    setNumberButton(R.id.button2, "2");
    setNumberButton(R.id.button3, "3");
    setNumberButton(R.id.button4, "4");
    setNumberButton(R.id.button5, "5");
    setNumberButton(R.id.button6, "6");
    setNumberButton(R.id.button7, "7");
    setNumberButton(R.id.button8, "8");
    setNumberButton(R.id.button9, "9");
```

```

// Operator buttons
setOperatorButton(R.id.buttonAdd, "+");
setOperatorButton(R.id.buttonSub, "-");
setOperatorButton(R.id.buttonMul, "*");
setOperatorButton(R.id.buttonDiv, "/");

// Equal button
Button equalButton = findViewById(R.id.buttonEqual);
equalButton.setOnClickListener(v -> {
    if (!currentInput.isEmpty()) {
        double secondOperand = Double.parseDouble(currentInput);
        double result = performOperation(firstOperand, secondOperand, operator);
        resultText.setText(String.valueOf(result));
        currentInput = "";
        operator = "";
    }
});

// Clear button
Button clearButton = findViewById(R.id.buttonClear);
clearButton.setOnClickListener(v -> {
    currentInput = "";
    operator = "";
    firstOperand = 0;
    resultText.setText("");
});
}

private void setNumberButton(int buttonId, String value) {
    Button button = findViewById(buttonId);
    button.setOnClickListener(v -> {
        if (isOperatorClicked) {
            currentInput = value;
            isOperatorClicked = false;
        } else {
            currentInput += value;
        }
        resultText.setText(currentInput);
    });
}

private void setOperatorButton(int buttonId, String op) {
    Button button = findViewById(buttonId);
    button.setOnClickListener(v -> {
        if (!currentInput.isEmpty()) {
            firstOperand = Double.parseDouble(currentInput);
            currentInput = "";
            operator = op;
            isOperatorClicked = true;
        }
    });
}

```

```
private double performOperation(double firstOperand, double secondOperand, String operator) {  
    switch (operator) {  
        case "+":  
            return firstOperand + secondOperand;  
        case "-":  
            return firstOperand - secondOperand;  
        case "*":  
            return firstOperand * secondOperand;  
        case "/":  
            if (secondOperand != 0) {  
                return firstOperand / secondOperand;  
            } else {  
                resultText.setText("Error");  
                return 0;  
            }  
        default:  
            return 0;  
    }  
}  
}
```

Assignment no:- 8

Demonstrate use of Scroll view in android?

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp">

    <!-- LinearLayout to hold all the content inside ScrollView -->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <!-- TextViews for demonstration -->
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="This is an example of ScrollView in Android."
            android:textSize="20sp"
            android:layout_marginBottom="16dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="You can place multiple UI elements inside a ScrollView."
            android:textSize="18sp"
            android:layout_marginBottom="16dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="When the content inside the ScrollView exceeds the screen size, the screen becomes
scrollable."
            android:textSize="18sp"
            android:layout_marginBottom="16dp" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="ScrollView allows you to add elements such as buttons, images, forms, etc."
            android:textSize="18sp"
            android:layout_marginBottom="16dp" />

        <TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Adding more content will make the ScrollView scrollable."
    android:textSize="18sp"
    android:layout_marginBottom="16dp" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="This is the final TextView to demonstrate scrolling."
    android:textSize="18sp"
    android:layout_marginBottom="16dp" /> </LinearLayout></ScrollView>
```

MainActivity.java (Logic)

```
package com.example.scrollviewdemo;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Set the layout to activity_main.xml
    }
}
```

Assignment no:- 9

Demonstrate use of intent in android?

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:id="@+id/buttonOpenSecondActivity"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Go to Second Activity"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

Define Layout for SecondActivity (activity_second.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textViewReceivedMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Received Message will be shown here"
        android:layout_centerInParent="true"
        android:textSize="18sp"/>

</RelativeLayout>
```

Implement MainActivity (MainActivity.java)

```
package com.example.intentdemo;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find the button by its ID
        Button buttonOpenSecondActivity = findViewById(R.id.buttonOpenSecondActivity);
```

```

// Set an onClick listener to handle the button click
buttonOpenSecondActivity.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Create an Intent to start SecondActivity
        Intent intent = new Intent(MainActivity.this, SecondActivity.class);

        // Add data to the Intent (sending a message)
        intent.putExtra("message", "Hello from MainActivity!");

        // Start SecondActivity
        startActivity(intent);
    }
});
}
}

```

Implement SecondActivity (SecondActivity.java)

```

package com.example.intentdemo;

import android.content.Intent;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class SecondActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second);

        // Find the TextView to display the received message
        TextView textViewReceivedMessage = findViewById(R.id.textViewReceivedMessage);

        // Get the Intent that started this activity
        Intent intent = getIntent();

        // Retrieve the message from the Intent
        String message = intent.getStringExtra("message");

        // Display the message in the TextView
        textViewReceivedMessage.setText(message);
    }
}

```

Register SecondActivity in AndroidManifest.xml

```

<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/Theme.IntentDemo">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

```



```
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<!-- Declare SecondActivity -->
<activity android:name=".SecondActivity"></activity>

</application>
```

Assignment no:- 10

Create application to demonstrate menu option?

Define the Layout (activity_main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Choose an option from the menu"
        android:textSize="20sp"
        android:layout_centerInParent="true"/>

</RelativeLayout>
```

MainActivity.java (Handle Menu Logic)

```
package com.example.menudemo;

import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Initialize TextView to show the menu choice
        textView = findViewById(R.id.textView);
    }

    // Inflate the menu; this will add items to the action bar if present.
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu options from res/menu/menu_main.xml
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true; // Return true to show the menu
    }

    // Handle item selection
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle clicks on the menu items
        switch (item.getItemId()) {
```

```

        case R.id.action_settings:
            textView.setText("Settings option selected");
            return true;
        case R.id.action_about:
            textView.setText("About option selected");
            return true;
        case R.id.action_exit:
            textView.setText("Exit option selected");
            finish(); // Close the app
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

```

Create Menu XML (res/menu/menu_main.xml)

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <!-- Settings option -->
    <item
        android:id="@+id/action_settings"
        android:title="Settings"
        android:icon="@android:drawable/ic_menu_preferences"
        android:showAsAction="ifRoom"/>

    <!-- About option -->
    <item
        android:id="@+id/action_about"
        android:title="About"
        android:icon="@android:drawable/ic_menu_info_details"
        android:showAsAction="ifRoom"/>

    <!-- Exit option -->
    <item
        android:id="@+id/action_exit"
        android:title="Exit"
        android:icon="@android:drawable/ic_menu_close_clear_cancel"
        android:showAsAction="never"/>

</menu>

```

Assignment no:-11

Create application to demonstrate progress bar?

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<!-- ProgressBar to show progress -->
    <ProgressBar
        android:id="@+id/progressBar"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="100dp"
        android:indeterminate="false"
        android:max="100"
        android:progress="0"
        android:visibility="gone" />

<!-- TextView to display progress -->
    <TextView
        android:id="@+id/progressText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Progress: 0%"
        android:textSize="18sp"
        android:layout_below="@id/progressBar"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp" />

<!-- Button to start progress -->
    <Button
        android:id="@+id/startButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start Progress"
        android:layout_below="@id/progressText"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="30dp"/>

</RelativeLayout>
```

MainActivity.java (Logic)

```
package com.example.progressbardemo;

import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;
```

```

public class MainActivity extends AppCompatActivity {
    private ProgressBar progressBar;
    private TextView progressText;
    private Button startButton;
    private int progress = 0;
    private Handler handler = new Handler(); // To update the progress in the UI thread
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Find views by their IDs
        progressBar = findViewById(R.id.progressBar);
        progressText = findViewById(R.id.progressText);
        startButton = findViewById(R.id.startButton);

        // Set a click listener for the start button
        startButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Start the progress simulation
                startProgress();
            }
        });
    }
    private void startProgress() {
        // Show the ProgressBar and TextView initially
        progressBar.setVisibility(View.VISIBLE);
        progressText.setVisibility(View.VISIBLE);

        // Disable the button to prevent multiple clicks while the task is in progress
        startButton.setEnabled(false);

        // Simulate progress with a handler and a runnable
        Runnable runnable = new Runnable() {
            @Override
            public void run() {
                // Increase the progress by 1 each time
                progress++;

                // Update the ProgressBar and TextView with the new progress
                progressBar.setProgress(progress);
                progressText.setText("Progress: " + progress + "%");

                // If the task is not finished, continue updating the progress
                if (progress < 100) {
                    handler.postDelayed(this, 50); // Call the runnable again after 50ms
                } else {
                    // When the task finishes, enable the button again
                    startButton.setEnabled(true);
                    progressText.setText("Task Complete!");
                }
            }
        };
    }
}

```

```
    }  
};  
  
    // Start the progress updates  
    handler.post(runnable);  
}  
}
```