

Android Studio Java Worksheets

Worksheet 1: Hello World App (Java)

Objective: Create and run a basic Android app that displays a message.

Steps:

1. Open Android Studio -> New Project -> Empty Activity
2. Project name: HelloWorldApp
3. Language: Java

4. In activity_main.xml, replace contents with:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/helloText"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, Android!"
        android:textSize="24sp"/>

</LinearLayout>
```

5. Run your app on an emulator or device

Challenge: Change the TextView text to include your name

Worksheet 2: Click Counter (Java)

Objective: Make a button that counts how many times it's clicked.

Steps:

1. New project -> Name: ClickCounterApp -> Language: Java

2. In activity_main.xml, use this layout:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:gravity="center"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/counterText"
        android:text="Count: 0"
        android:textSize="24sp"
        android:layout_marginBottom="20dp"/>

    <Button
        android:id="@+id/countButton"
```

Android Studio Java Worksheets

```
        android:text="Click Me!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"/>
</LinearLayout>
```

3. In MainActivity.java, add:

```
int count = 0;
TextView counterText = findViewById(R.id.counterText);
Button countButton = findViewById(R.id.countButton);
countButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        count++;
        counterText.setText("Count: " + count);
    }
});
```

Challenge: Add a "Reset" button to set the count back to 0

Worksheet 3: Simple Todo List (Java)

Objective: Create a basic to-do list using RecyclerView.

Steps:

1. Create a new project: SimpleTodoList (Java)

2. Add a RecyclerView, EditText, and Button to activity_main.xml:

```
<EditText android:id="@+id/editTextTask" android:hint="Enter task" />
<Button android:id="@+id/addButton" android:text="Add Task"/>
<androidx.recyclerview.widget.RecyclerView android:id="@+id/recyclerView"/>
```

3. Create:

- TodoAdapter.java (extends RecyclerView.Adapter)
- TaskViewHolder.java
- task_item.xml for individual items

4. In MainActivity.java:

- Create a list to store tasks
- Set up RecyclerView and Adapter
- Add items on button click

Optional: Add a long-click listener in the adapter to delete tasks

Challenge: Save and restore the list using SharedPreferences

Worksheet 4: Navigation with Intents (Java)

Objective: Learn to switch between two screens using Intents.

Android Studio Java Worksheets

Steps:

1. Create a new project: TwoScreenApp -> Java

2. Add a button to activity_main.xml:

```
<Button android:id="@+id/navigateButton" android:text="Go to Second Screen"/>
```

3. Create a second activity: SecondActivity.java

4. In MainActivity.java:

```
Intent intent = new Intent(MainActivity.this, SecondActivity.class);
startActivity(intent);
```

5. In SecondActivity.java: show "Welcome to the second screen!"

Challenge: Pass a string using Intent.putExtra

Worksheet 5: Save User Input with SharedPreferences (Java)

Objective: Save and load text input between app sessions.

Steps:

1. Create a new project: UserNoteSaver -> Java

2. In activity_main.xml:

```
<EditText android:id="@+id/inputNote" android:hint="Enter your note"/>
<Button android:id="@+id/saveButton" android:text="Save Note"/>
<TextView android:id="@+id/noteDisplay"/>
```

3. In MainActivity.java:

```
SharedPreferences sharedPreferences = getSharedPreferences("MyPrefs", MODE_PRIVATE);
Load saved note and display it
```

```
saveButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String note = inputNote.getText().toString();
        SharedPreferences.Editor editor = sharedPreferences.edit();
        editor.putString("note", note);
        editor.apply();
        noteDisplay.setText(note);
    }
});
```

Challenge: Add a "Clear" button to delete the saved note