

Qiong LIU

Activity related codes

0.1 Java code

Below is an example of a default Java class for an Android activity.

```
package com.example.myapplication;
// Package declaration for the application.

import android.os.Bundle;
// Import for managing activity state using the Bundle class.

import androidx.activity.EdgeToEdge;
// Enable edge-to-edge display, i.e., full screen.

import androidx.appcompat.app.AppCompatActivity;
// Manage System UI Interactions, Dynamic Layout Adjustments

import androidx.core.graphics.Insets;
// Handles inset values for system bars such as status and navigation bars.

import androidx.core.view.ViewCompat;
// Used to work with views in a backward-compatible way.

import androidx.core.view.WindowInsetsCompat;
// Provides information about system windows and their insets.

public class MainActivity extends AppCompatActivity {
    // Main class extending AppCompatActivity for compatibility across different Android versions

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // onCreate method, called when the activity is first created.
        //A Bundle is a data structure that stores key-value pairs.
        //When the activity is first created, savedInstanceState is null; If the activity is being recreated after
        //being previously destroyed, savedInstanceState contains the data saved in the onSaveInstanceState()
        //method.

        super.onCreate(savedInstanceState);
        // Call the superclass's onCreate method to initialize the activity.

        EdgeToEdge.enable(this);
        // Enable edge-to-edge mode, allowing the app to use the full screen

        setContentView(R.layout.activity_main);
        // Set the content view to the layout file 'activity_main.xml'.

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            // Set an OnApplyWindowInsetsListener to adjust the view when system window insets (e.g., status bar,
            // navigation bar) are applied.

            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            // Get the insets for system bars (status bar, navigation bar) using the WindowInsetsCompat API.

            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            // Set padding for the view based on the system bar insets, ensuring the content is adjusted to avoid
            // being overlapped.

            return insets;
        });
    }
}
```

0.2 Android XML Code Breakdown

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Root layout begins, using ConstraintLayout -->
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
```

```

android:id="@+id/main" <!-- Assign an ID to the layout for reference in code -->
android:layout_width="match_parent" <!-- Layout width fills the parent container -->
android:layout_height="match_parent" <!-- Layout height fills the parent container -->
tools:context=".MainActivity"> <!-- Specifies that this layout is associated with MainActivity -->

<TextView
    android:layout_width="wrap_content" <!-- Width of TextView only as wide as its content -->
    android:layout_height="wrap_content" <!-- Height of TextView only as tall as its content -->
    android:text="Hello World!" <!-- Text displayed in TextView -->
    app:layout_constraintBottom_toBottomOf="parent" <!-- Align the bottom of TextView with the bottom of its parent -->
    app:layout_constraintEnd_toEndOf="parent" <!-- Align the end (right) of TextView with the end of its parent -->
    app:layout_constraintStart_toStartOf="parent" <!-- Align the start (left) of TextView with the start of its parent -->
    app:layout_constraintTop_toTopOf="parent" /> <!-- Align the top of TextView with the top of its parent -->

</androidx.constraintlayout.widget.ConstraintLayout> <!-- Root layout ends -->

```

0.3 Exercise 1

Create a simple Android application that includes a main activity (MainActivity) and an XML layout file. The app should contain a button and a text view (TextView). When the button is clicked, the text view should update to display the current date and time.

```

<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Show Date and Time"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

<TextView
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Date and time will be shown here"
    app:layout_constraintTop_toBottomOf="@id/button"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />

```

```

public class MainActivity extends AppCompatActivity {

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

        final Button button = findViewById(R.id.button);
        final TextView textView = findViewById(R.id.textView);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String currentDateTime = DateFormat.getDateTimeInstance().format(new Date());
                textView.setText(currentDateTime);
            }
        });
    }
}

```

0.4 Exercise 2

Create a basic Android application where the user can enter a note into a text field. The application will save the text when the screen is rotated and restore it once the rotation is complete, preventing any data loss.

```
package com.example.cml;

import android.os.Bundle;
import android.widget.EditText;
import android.widget.TextView;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity2 extends AppCompatActivity {

    private EditText editTextNote;
    private TextView textViewSavedNote;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main2);

        editTextNote = findViewById(R.id.editTextNote);
        textViewSavedNote = findViewById(R.id.textViewSavedNote);
        if (savedInstanceState != null) {
            String savedText = savedInstanceState.getString("NOTE");
            editTextNote.setText(savedText);
            textViewSavedNote.setText(savedText);
        }

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main2), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }

    @Override
    protected void onSaveInstanceState(Bundle outState) {
        super.onSaveInstanceState(outState);
        outState.putString("NOTE", editTextNote.getText().toString());
    }

    @Override
    protected void onRestoreInstanceState(Bundle savedInstanceState) {
        super.onRestoreInstanceState(savedInstanceState);
        String savedText = savedInstanceState.getString("NOTE");
        editTextNote.setText(savedText);
        textViewSavedNote.setText(savedText);
    }
}
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:id="@+id/main2"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity2">

    <!-- EditText for entering text -->
    <EditText
        android:id="@+id/editTextNote"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```

        android:hint="Enter your note here"
        android:inputType="text"
        android:gravity="start" />

<!-- TextView to display the saved note -->
<TextView
    android:id="@+id/textViewSavedNote"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:layout_marginTop="20dp"
    android:text="Saved note will appear here"
    android:background="#ECECEC"
    android:padding="10dp" />

</LinearLayout>

```

0.5 Exercise 3

Create an Android app that allows the user to enter a message in the MainActivity, which is then passed to the SecondActivity and displayed there.

```

public class MainActivity3 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main3);

        final EditText editTextMessage = findViewById(R.id.editTextMessage);
        Button buttonSend = findViewById(R.id.buttonSend);

        buttonSend.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String message = editTextMessage.getText().toString();
                Intent intent = new Intent(MainActivity3.this, MainActivity4.class);
                intent.putExtra("MESSAGE", message);
                startActivity(intent);
            }
        });
    }
}

```

```

public class MainActivity4 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main4);

        TextView textViewMessage = findViewById(R.id.textViewMessage);
        String message = getIntent().getStringExtra("MESSAGE");
        textViewMessage.setText(message);

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
            Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
            return insets;
        });
    }
}

```

```

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"

```

```

        android:orientation="vertical"
        android:id="@+id/main3"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="16dp"
        android:gravity="center_vertical"
        tools:context=".MainActivity3">

        <EditText
            android:id="@+id/editTextMessage"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="Enter a message" />

        <Button
            android:id="@+id/buttonSend"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center_horizontal"
            android:text="Send Message" />
    </LinearLayout>

```

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000"
    tools:context=".MainActivity4">

    <TextView
        android:id="@+id/textViewMessage"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Date and time will be shown here"
        android:textSize="32sp"
        android:gravity="center"
        android:textColor="#C2B280"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

```