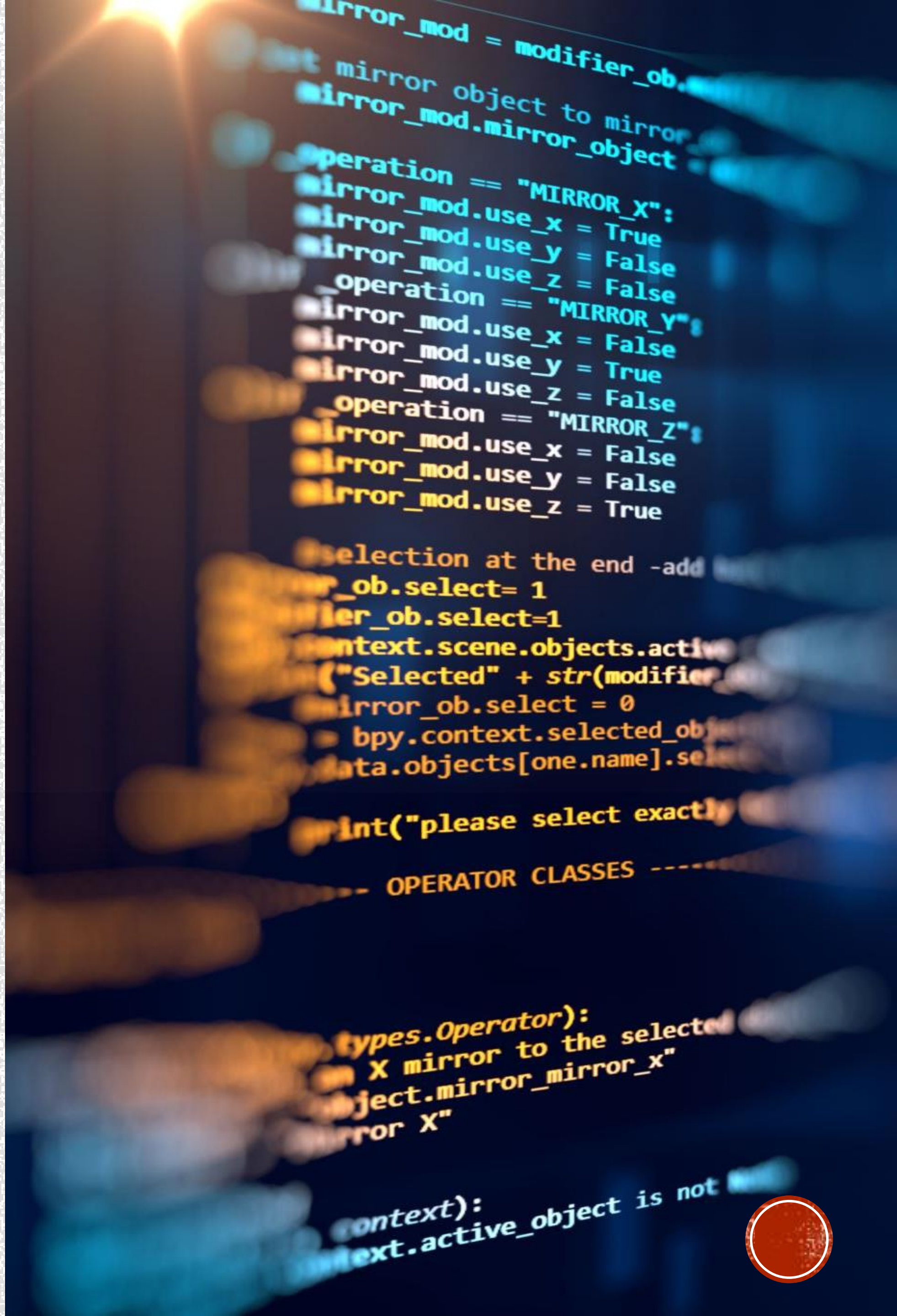
LECTURE 08:

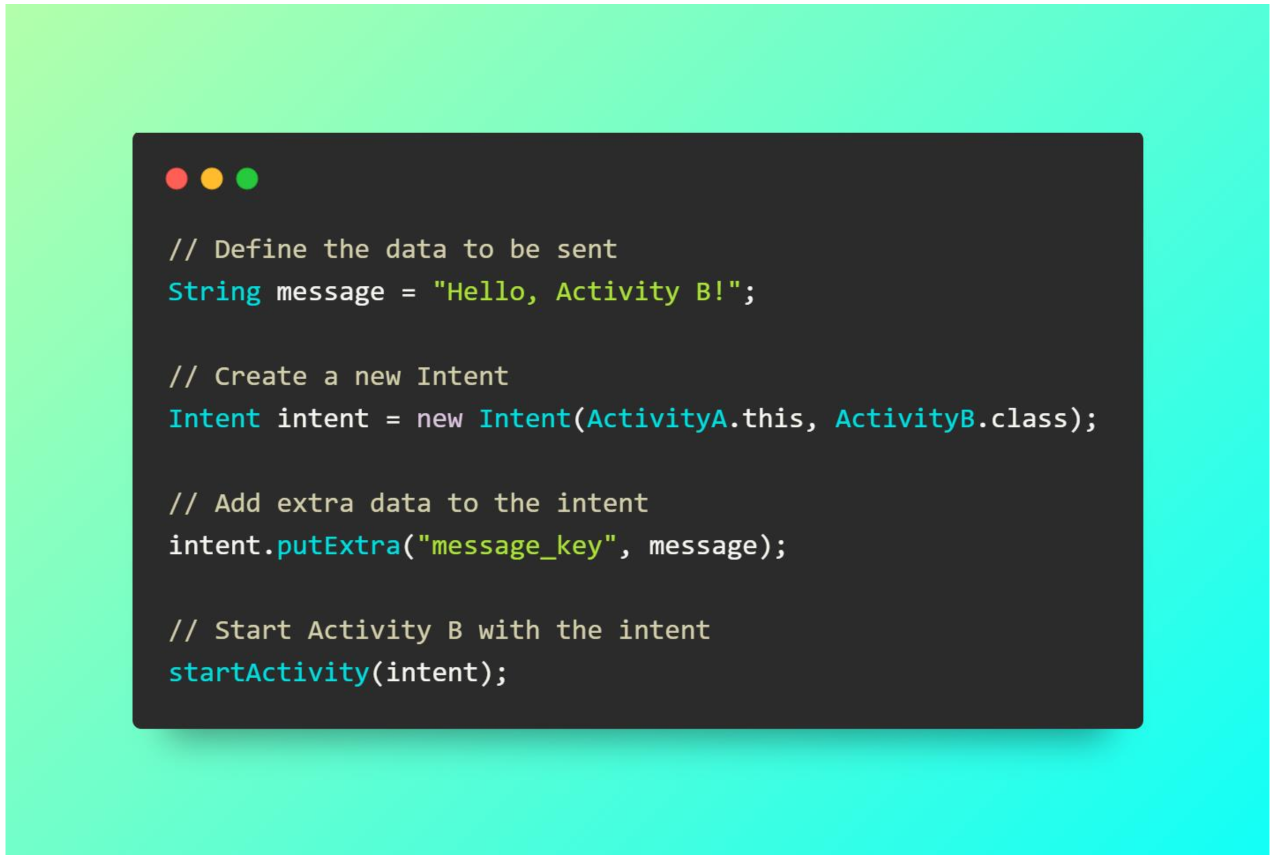
COMMUNICATION BETWEEN ACTIVITIES: SENDING DATA & SNACKBAR IN ANDROID UI

COMMUNICATION BETWEEN INTENTS:

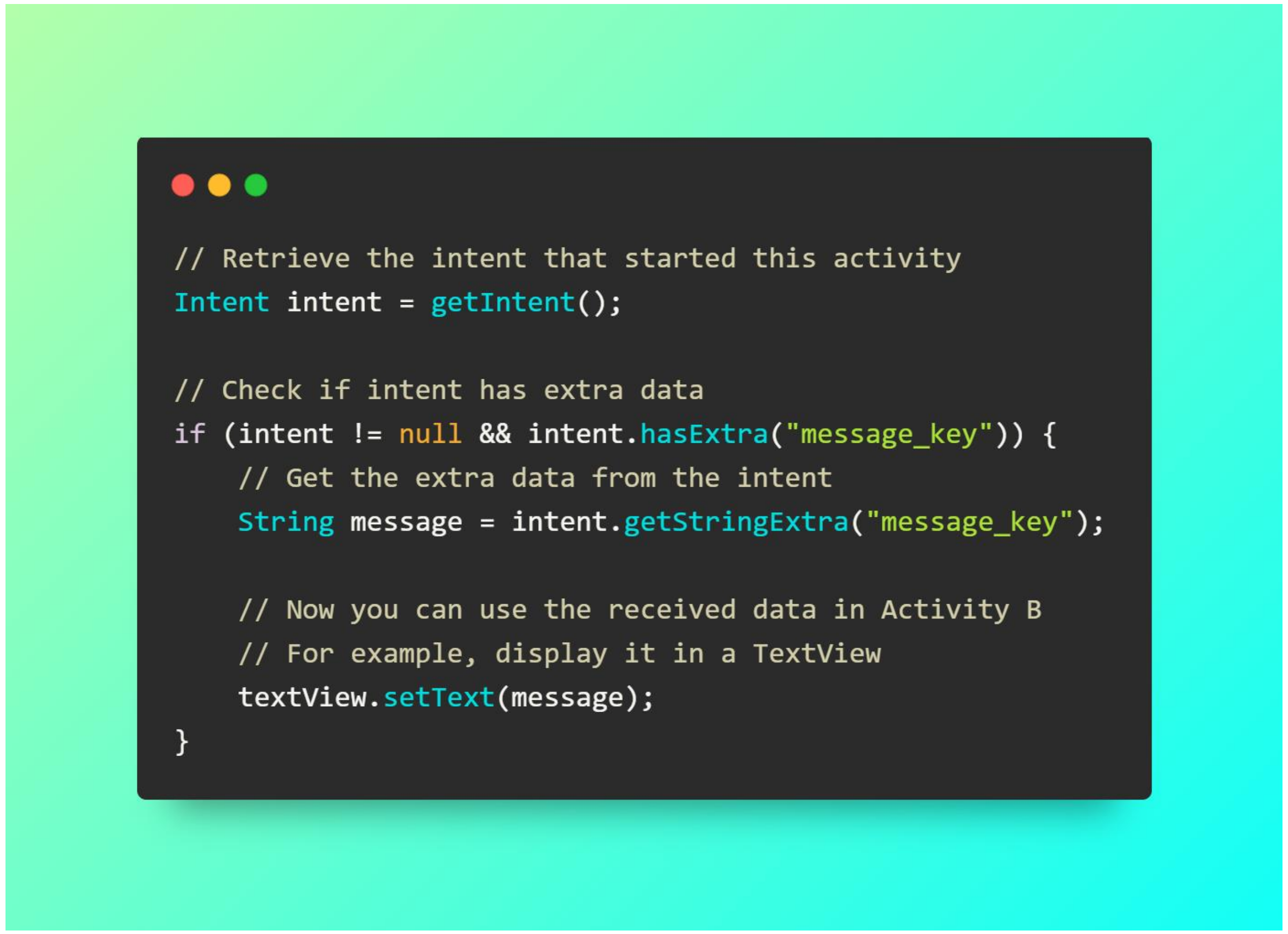
- Communication between activities in the context of Android development refers to the exchange of data, instructions, or signals between different screens or user interface components within an Android application. Activities are the building blocks of an Android app, representing individual screens with which users interact.
- When we talk about communication between activities, it typically involves passing data from one activity to another, triggering actions in one activity based on user interactions in another, or navigating between different activities in response to user actions or system events. This communication is facilitated using a mechanism called intents, which act as messengers to convey information between activities.



SENDING DATA USING INTENT:


```
// Define the data to be sent
String message = "Hello, Activity B!";

// Create a new Intent
Intent intent = new Intent(ActivityA.this, ActivityB.class);

// Add extra data to the intent
intent.putExtra("message_key", message);

// Start Activity B with the intent
startActivity(intent);
```

RECEIVING DATA FROM INTENTS:

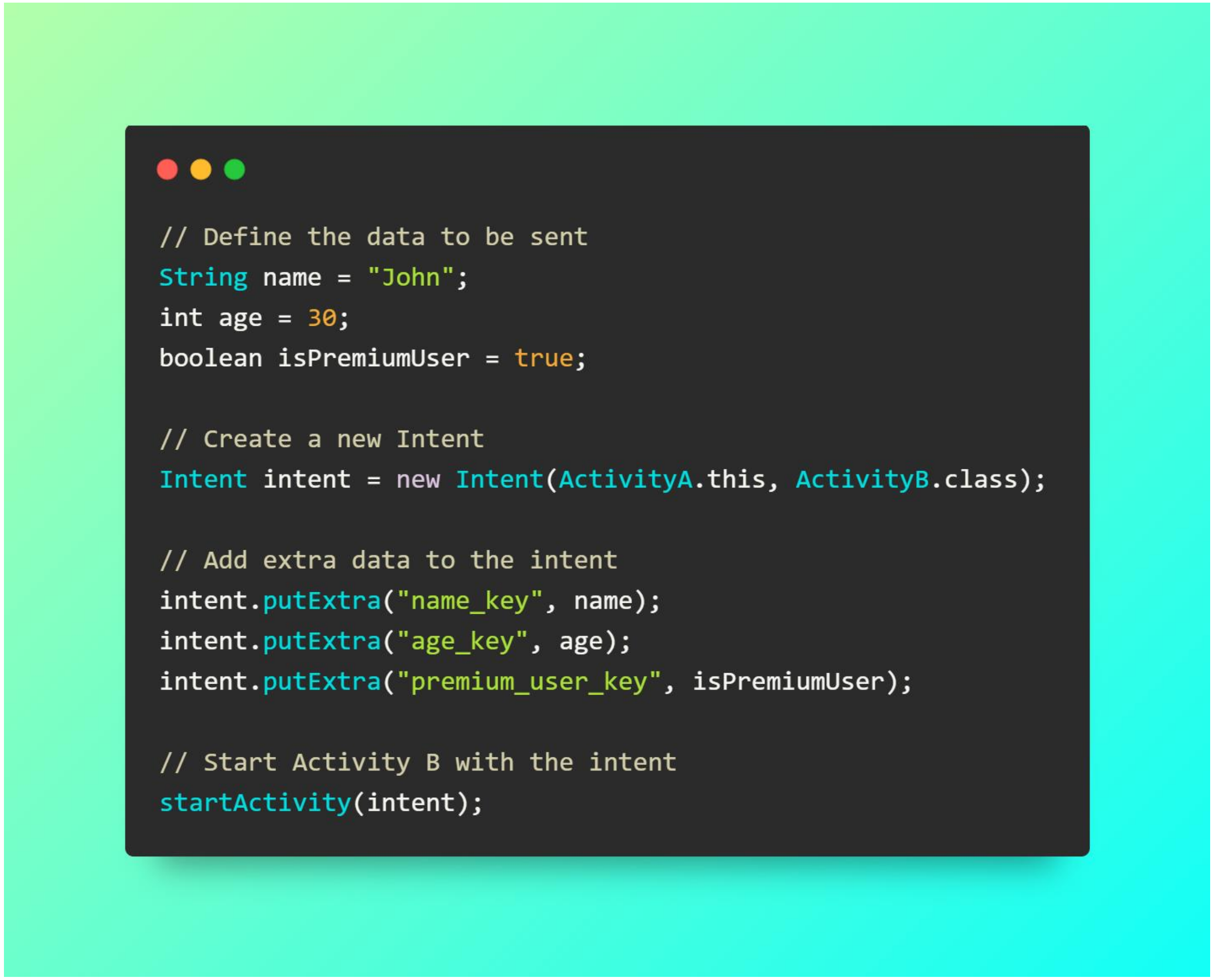


```java
// Retrieve the intent that started this activity
Intent intent = getIntent();

// Check if intent has extra data
if (intent != null && intent.hasExtra("message_key")) {
    // Get the extra data from the intent
    String message = intent.getStringExtra("message_key");

    // Now you can use the received data in Activity B
    // For example, display it in a TextView
    textView.setText(message);
}
```

EXPLANATION:

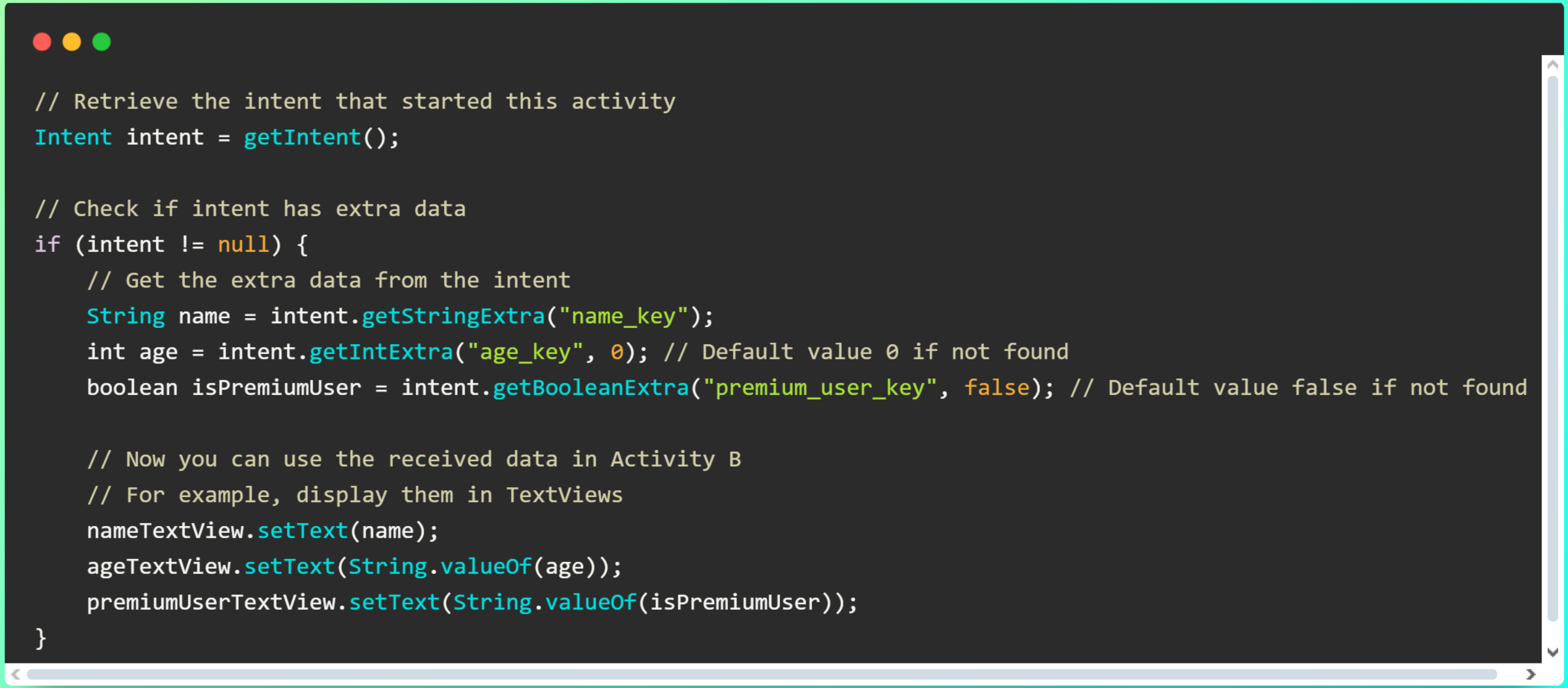
- In Activity A, we create an Intent and add extra data to it using the **putExtra() method**. We pass a key-value pair, where "**message_key**" is the key and "Hello, Activity B!" is the value.
- Then, we start Activity B by calling startActivity(intent).
- In Activity B, we retrieve the Intent that started this activity using **getIntent().**
- We then **check** if the intent has extra data using hasExtra().
- If **extra data is present**, we extract the data using getStringExtra() by providing the key ("message_key").
- Finally, we can use the received data (in this case, the message) as needed in Activity B, such as displaying it in a TextView.

SENDING MULTIPLE DATA:



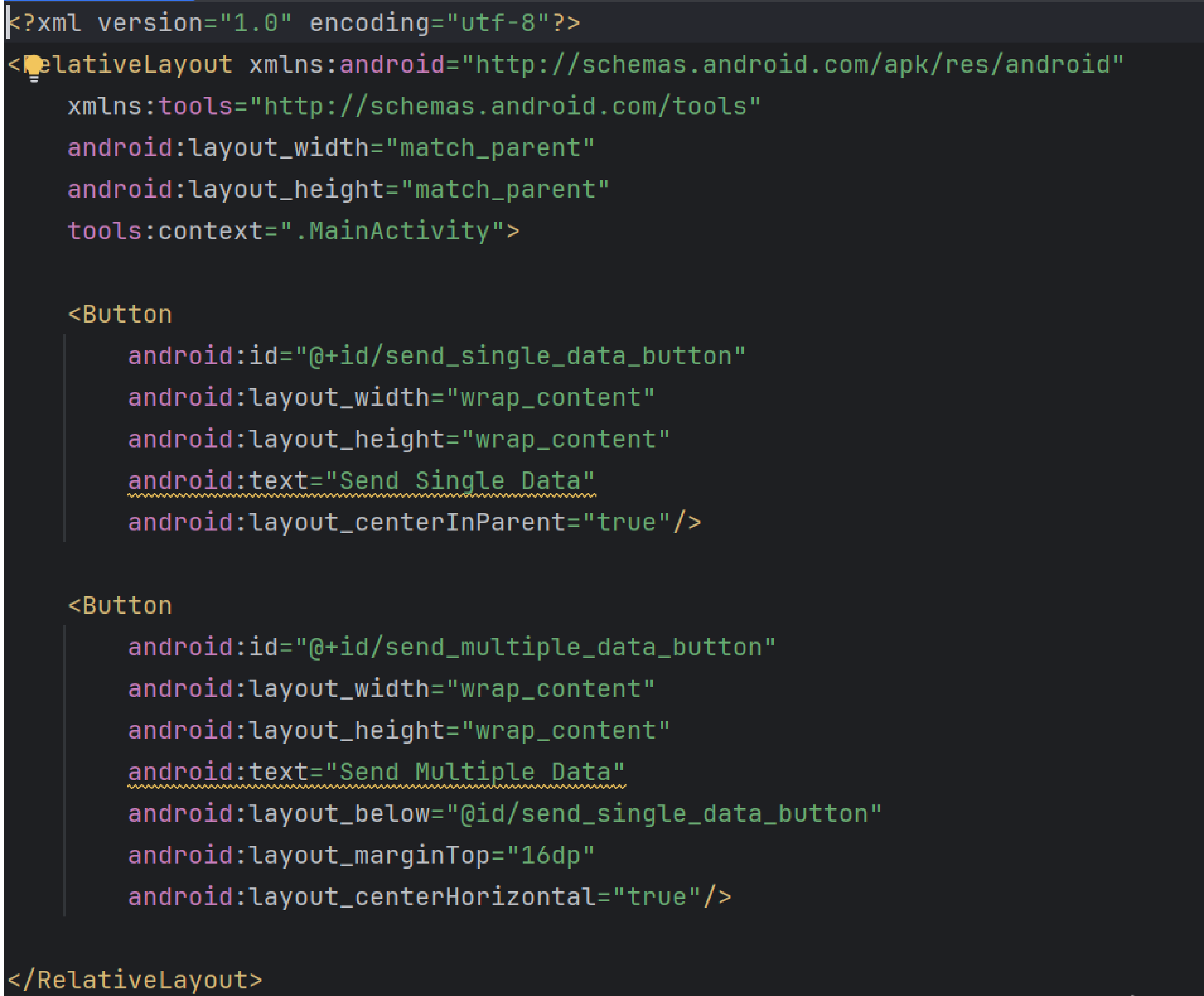
```java
// Define the data to be sent
String name = "John";
int age = 30;
boolean isPremiumUser = true;

// Create a new Intent
Intent intent = new Intent(ActivityA.this, ActivityB.class);

// Add extra data to the intent
intent.putExtra("name_key", name);
intent.putExtra("age_key", age);
intent.putExtra("premium_user_key", isPremiumUser);

// Start Activity B with the intent
startActivity(intent);
```

RECEIVING MULTIPLE DATA:



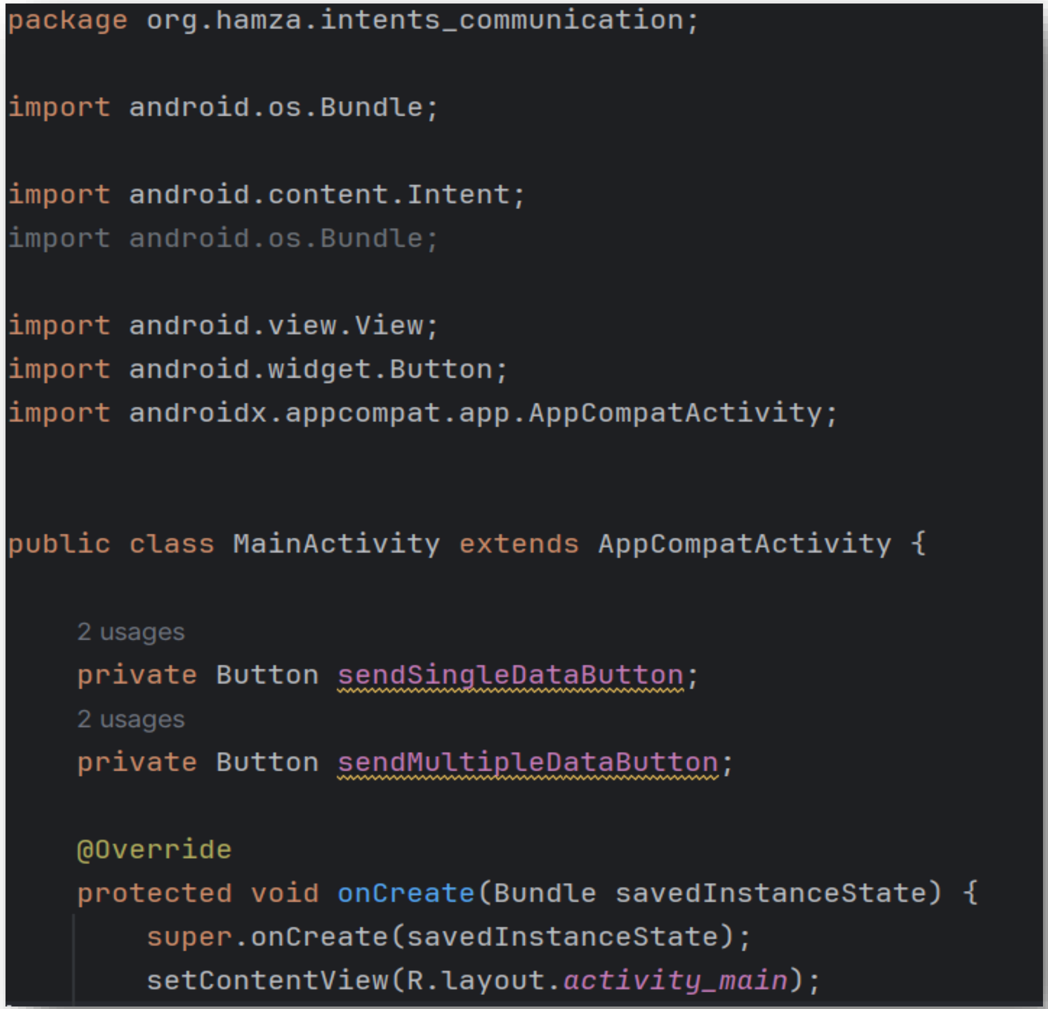
```java
// Retrieve the intent that started this activity
Intent intent = getIntent();

// Check if intent has extra data
if (intent != null) {
    // Get the extra data from the intent
    String name = intent.getStringExtra("name_key");
    int age = intent.getIntExtra("age_key", 0); // Default value 0 if not found
    boolean isPremiumUser = intent.getBooleanExtra("premium_user_key", false); // Default value false if not found

    // Now you can use the received data in Activity B
    // For example, display them in TextViews
    nameTextView.setText(name);
    ageTextView.setText(String.valueOf(age));
    premiumUserTextView.setText(String.valueOf(isPremiumUser));
}
```

EXAMPLE CODE FIRST ACTIVITY_XML: (SENDING DATA)



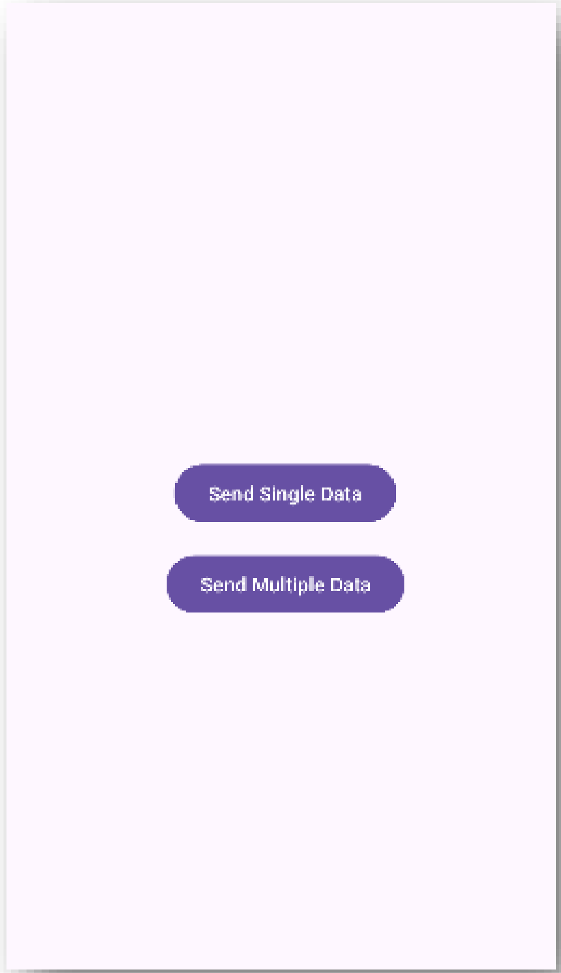
```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/send_single_data_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Single Data"
        android:layout_centerInParent="true"/>

    <Button
        android:id="@+id/send_multiple_data_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Send Multiple Data"
        android:layout_below="@id/send_single_data_button"
        android:layout_marginTop="16dp"
        android:layout_centerHorizontal="true"/>

</RelativeLayout>
```

MAIN.JAVA FILE:

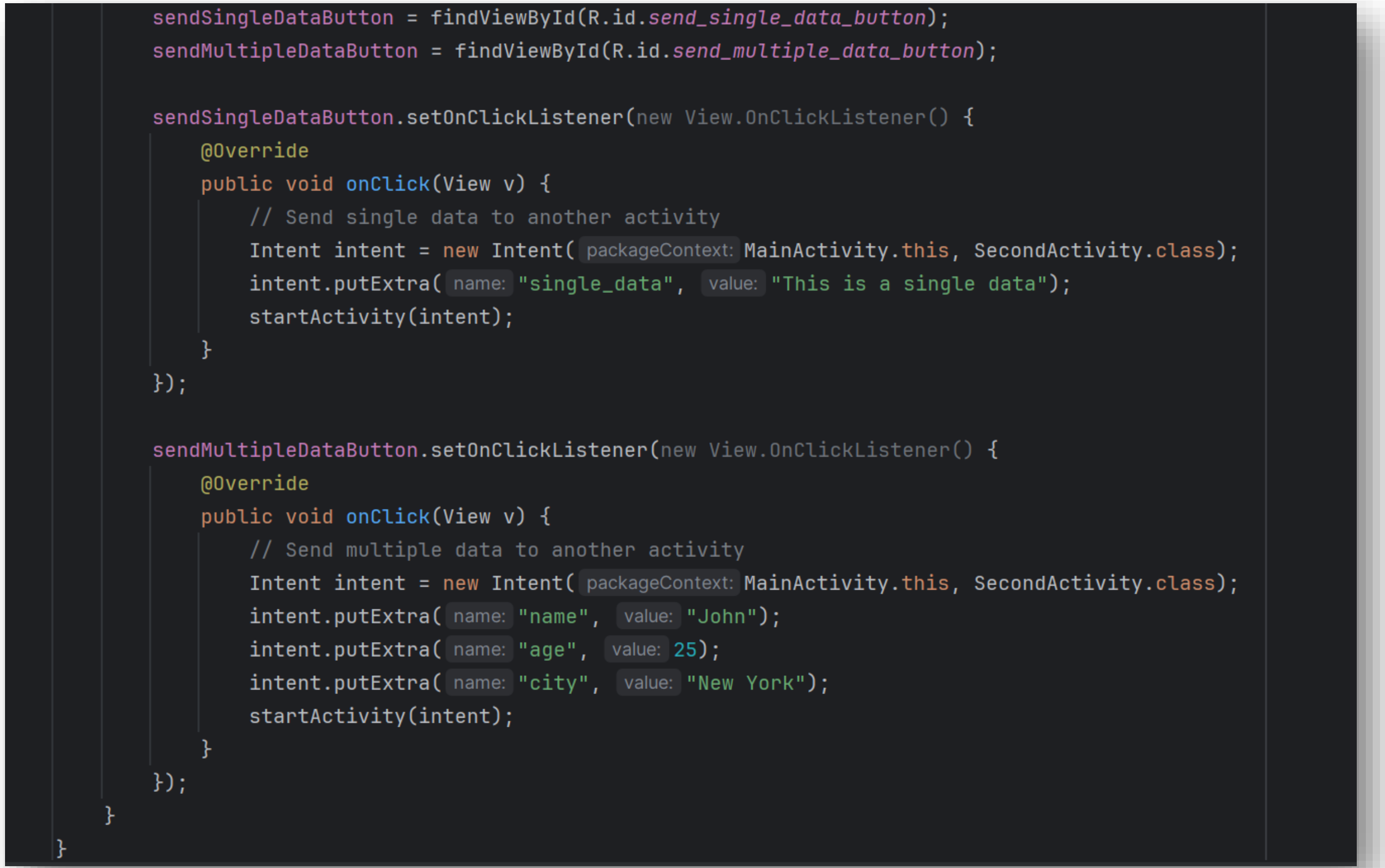


```java
package org.hamza.intents_communication;

import android.os.Bundle;

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 {

    2 usages
    private Button sendSingleDataButton;
    2 usages
    private Button sendMultipleDataButton;

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

FIRST
ACTIVITY_XML: OUTPUT

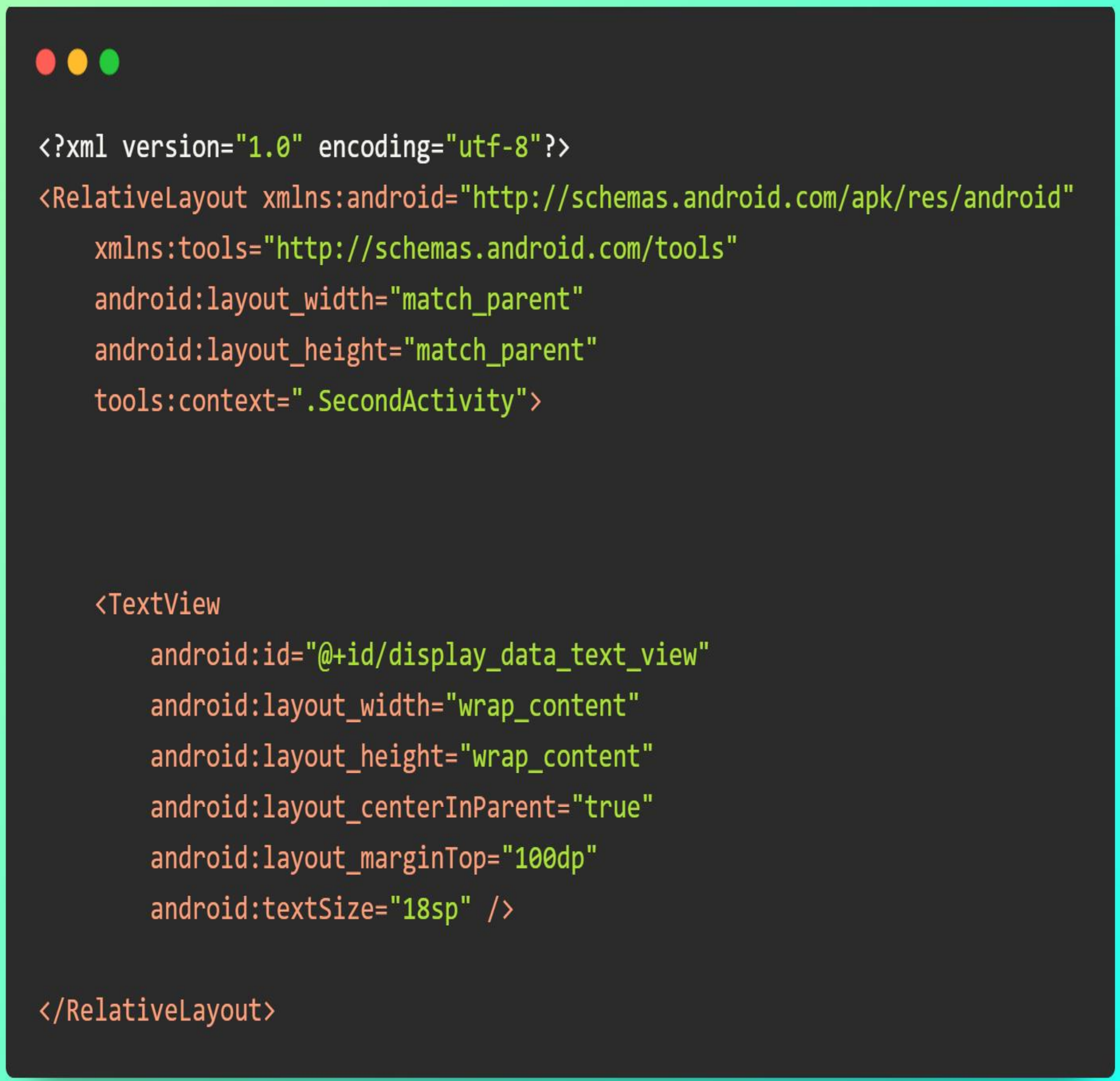


FIRST ACTIVITY JAVA FILE:



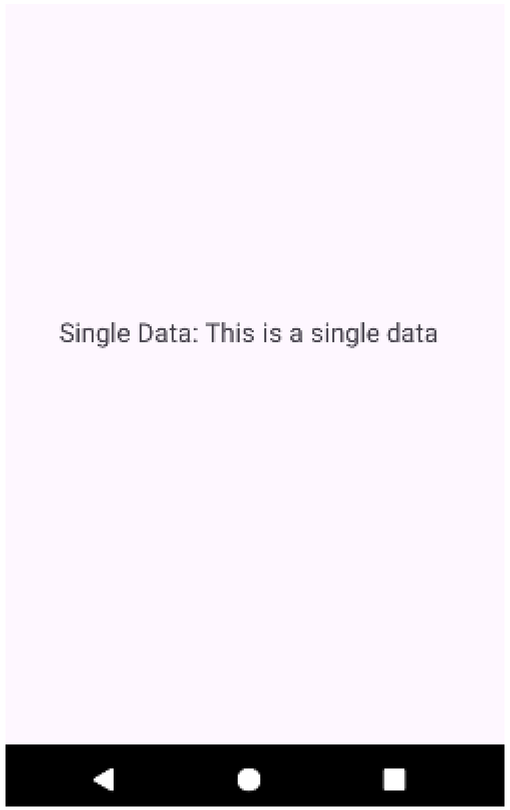
```java
        sendSingleDataButton = findViewById(R.id.send_single_data_button);
        sendMultipleDataButton = findViewById(R.id.send_multiple_data_button);

        sendSingleDataButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Send single data to another activity
                Intent intent = new Intent( packageContext: MainActivity.this, SecondActivity.class);
                intent.putExtra( name: "single_data",  value: "This is a single data");
                startActivity(intent);
            }
        });

        sendMultipleDataButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Send multiple data to another activity
                Intent intent = new Intent( packageContext: MainActivity.this, SecondActivity.class);
                intent.putExtra( name: "name",  value: "John");
                intent.putExtra( name: "age",  value: 25);
                intent.putExtra( name: "city",  value: "New York");
                startActivity(intent);
            }
        });
    }
}
```

SECOND ACTIVITY XML:

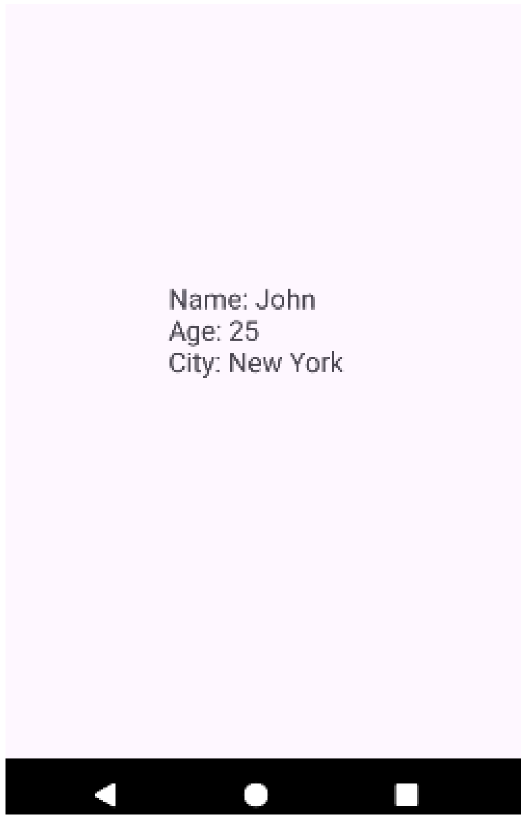


```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity">



    <TextView
        android:id="@+id/display_data_text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:layout_marginTop="100dp"
        android:textSize="18sp" />

</RelativeLayout>
```

SECOND ACTIVITY XML: OUTPUT

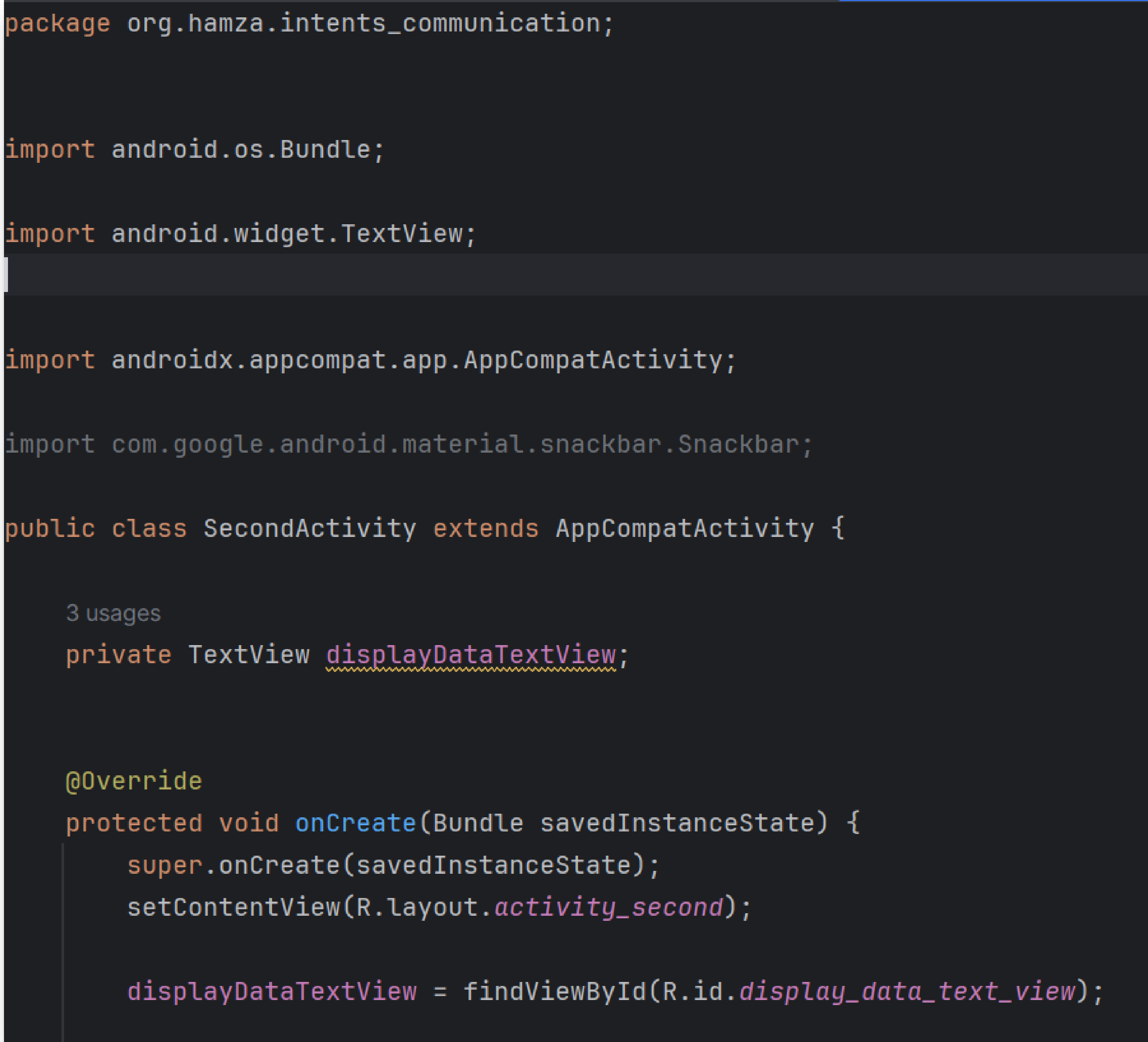


```
Single Data: This is a single data
```



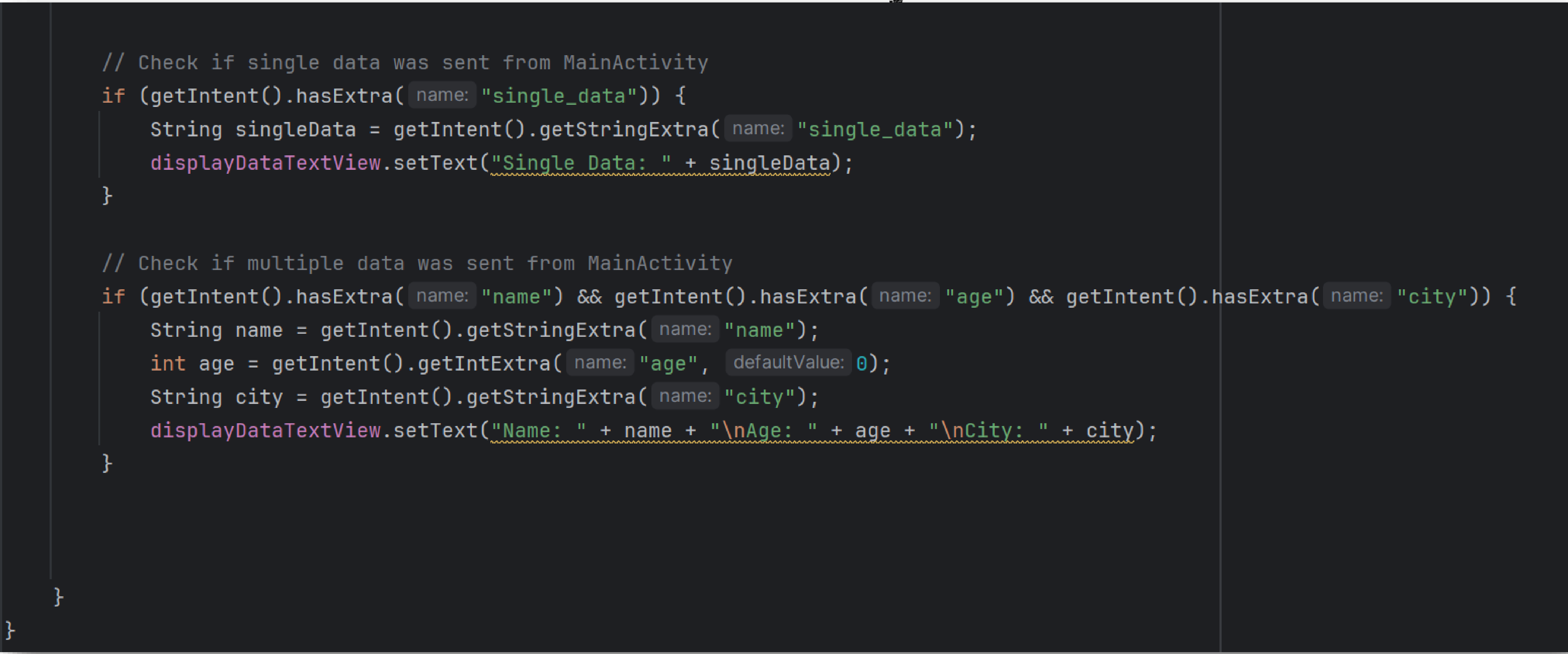
```
Name: John
Age: 25
City: New York
```

SECOND ACTIVITY JAVA FILE:



```
package org.hamza.intents_communication;


import android.os.Bundle;

import android.widget.TextView;


import androidx.appcompat.app.AppCompatActivity;

import com.google.android.material.snackbar.Snackbar;

public class SecondActivity extends AppCompatActivity {

    3 usages
    private TextView displayDataTextView;


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

        displayDataTextView = findViewById(R.id.display_data_text_view);
```

SECOND ACTIVITY JAVA FILE:



```java
        // Check if single data was sent from MainActivity
        if (getIntent().hasExtra( name: "single_data")) {
            String singleData = getIntent().getStringExtra( name: "single_data");
            displayDataTextView.setText("Single Data: " + singleData);
        }

        // Check if multiple data was sent from MainActivity
        if (getIntent().hasExtra( name: "name") && getIntent().hasExtra( name: "age") && getIntent().hasExtra( name: "city")) {
            String name = getIntent().getStringExtra( name: "name");
            int age = getIntent().getIntExtra( name: "age",  defaultValue: 0);
            String city = getIntent().getStringExtra( name: "city");
            displayDataTextView.setText("Name: " + name + "\nAge: " + age + "\nCity: " + city);
        }


    }
}
```

TASK 1:

Task: Implement Login Functionality with Communication Between Activities

Create an Android app that allows users to log in using a username and password. Upon successful login, the app should navigate to a welcome screen displaying a personalized welcome message with the user's name.

1. **Design the Login Form UI:**
   - Create a layout file (activity_login.xml) for the login screen with EditTexts for entering username and password, and a Button for logging in.
   - Add any additional UI elements like TextViews for error messages or links to registration or forgot password screens.
2. **Implement Login Functionality:**
   - In LoginActivity, implement logic to validate the username and password entered by the user.
   - Validate the input fields to ensure they are not empty and meet any required criteria (e.g., minimum length).
   - Check the entered username and password against predefined values
   - If the credentials are valid, proceed to the welcome screen. Otherwise, display appropriate error messages to the user.

CONTINUE:

3.Communicate Between Activities:

•Once the user successfully logs in, pass the username or user object to the welcome screen activity (WelcomeActivity) using intents.
•Create an intent in LoginActivity to start WelcomeActivity.
•Add the username or user object as an extra to the intent.
•Start WelcomeActivity using startActivity(intent).

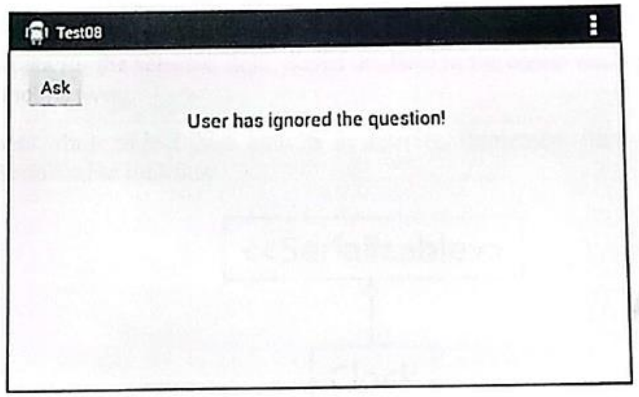
4.Display Welcome Message:

•In WelcomeActivity, retrieve the username or user object passed from LoginActivity using getIntent().
•Extract the username or user details from the intent extras.
•Display a personalized welcome message on the welcome screen, incorporating the username or user details.
•You can use a TextView or any other UI element to display the welcome message.

TASK 2:

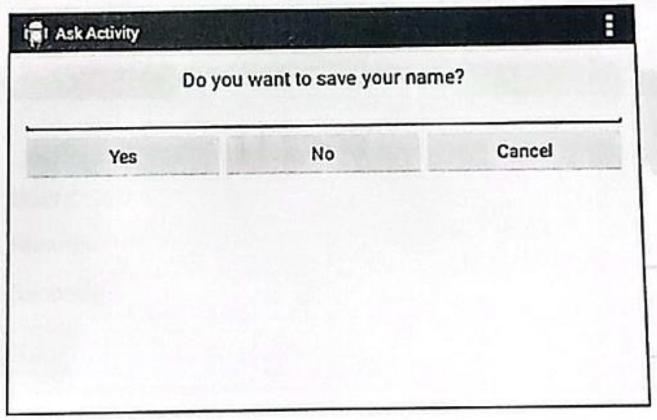
Create an application which has the following activities:

a) MainActivity



User has ignored the question!

b) AskActivity



Do you want to save your name?

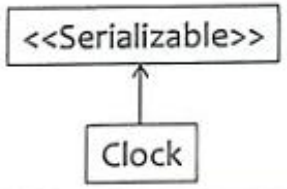
TASK 3:

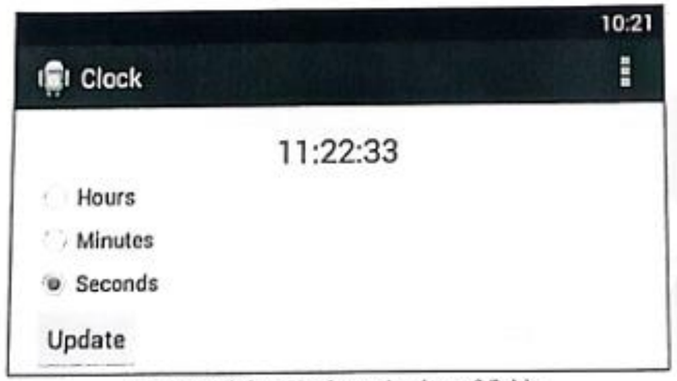
Create an application which can manipulate a Clock object having hours, minutes and seconds as integer attributes. Your application must have only two activities as follows:

1) *ClockActivity*
   Shows the clock contents and the gives field selection by radio buttons for update. Creates new Clock object, displays it and send it to UpdateActivity for update.

2) *UpdateActivity*
   Gets data for the selected field. Saves the field in the object and send back object to ClockActivity.

You must pass whole object from activity to activity. Remember that your class must implement Serializable interface.

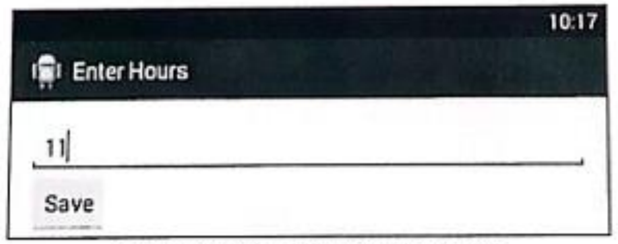




11:22:33

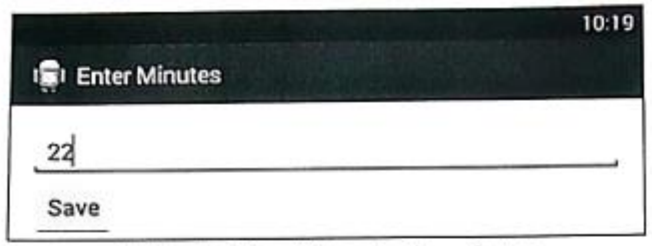
Hours
Minutes
[✓ Seconds]

ClockActivity: Having selection of fields



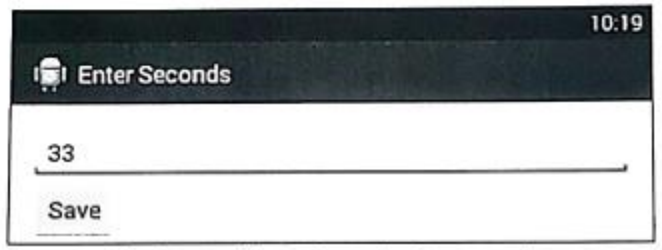
11

UpdateActivity: After selecting hours



22

UpdateActivity: After selecting minutes



33

UpdateActivity: After selecting seconds