

Exam

Exercise 1 : (10 Points)

Read each question carefully and choose the best option from the given choices (A, B, C, or D). Select the option that you think is correct and mark it on your answer sheet. Choose the option that most accurately answers the question.

- Q1: What is the purpose of the AndroidManifest.xml file in an Android project?
- Q2: In Android development, which XML-based file is used to define the user interface of an activity?
- Q3: What is the purpose of an Intent in Android?
- Q4: Which Android component is used to perform long-running operations in the background without blocking the user interface?
- Q5: Which of the following permissions is required to access the device's camera in an Android app?
- Q6: Which layout manager arranges the child views in a grid-like format in Android?
- Q7: Which of the following components is responsible for managing the user interface of an Android application?
- Q8: Which lifecycle method is called when an activity is no longer visible to the user?
- Q9: What is the purpose of the Gradle build system in Android development?
- Q10: Which of the following data storage options is suitable for storing large amounts of structured data in an Android app?

Question	Option A	Option B	Option C	Option D
Q1	Define app permissions	Declare app components	Specify app metadata	All of the above
Q2	activity_main.xml	manifest.xml	layout.xml	ui.xml
Q3	Data storage	User input handling	Component communication	Background processing
Q4	Activity	Service	Broadcast Receiver	Content Provider
Q5	CAMERA	INTERNET	READ_EXTERNAL_STORAGE	ACCESS_FINE_LOCATION
Q6	LinearLayout	RelativeLayout	GridLayout	FrameLayout
Q7	Activity	Service	Fragment	BroadcastReceiver
Q8	onCreate()	onPause()	onResume()	onDestroy()
Q9	Code compilation	Dependency management	Resource merging	All of the above
Q10	SharedPreferences	SQLite database	Internal storage	Network storage

Exercise 2 : (10 Points)

Create a Decimal to Binary conversion activity in Android Studio using Java and XML. The activity should have an input field for the user to enter a decimal number and a "Convert" button. When the user clicks the "Convert" button, the decimal number should be converted to binary and displayed in a TextView.

Requirements:**1. XML Layout:**

- * Create an XML layout file for the conversion activity.
- * Include an EditText view for the user to enter a decimal number.
- * Add a Button view for the "Convert" button.
- * Include a TextView view to display the binary result.

2. Java Code:

- * Create a Java class for the conversion activity.
- * Associate the XML layout with the Java class using setContentView() method.
- * Implement a click listener for the "Convert" button.
- * Convert the decimal number entered by the user to binary using the Integer.toBinaryString() method.
- * Display the binary result in the TextView.

3. Validation:

- * Perform validation to ensure that the input is a valid decimal number.
- * Display an error message if the input is invalid.

GOOD LUCK.
DR. MEFTAH ZOUAI

Model answer for the exam

Exercise 1 : (10 Points)

Read each question carefully and choose the best option from the given choices (A, B, C, or D). Select the option that you think is correct and mark it on your answer sheet. Choose the option that most accurately answers the question.

- Q1: Which of the following components is responsible for managing the user interface of an Android application?
- Q2: In Android development, which XML-based file is used to define the user interface of an activity?
- Q3: Which Android component is used to perform long-running operations in the background without blocking the user interface?
- Q4: What is the purpose of an Intent in Android?
- Q5: Which of the following permissions is required to access the device's camera in an Android app?
- Q6: What is the purpose of the AndroidManifest.xml file in an Android project?
- Q7: Which lifecycle method is called when an activity is no longer visible to the user?
- Q8: Which layout manager arranges the child views in a grid-like format in Android?
- Q9: What is the purpose of the Gradle build system in Android development?
- Q10: Which of the following data storage options is suitable for storing large amounts of structured data in an Android app?

Q	Option A	Option B	Option C	Option D	Correct Answer
1	Activity	Service	BroadcastReceiver	ContentProvider	A
2	Styles.xml	Manifest.xml	Strings.xml	Layout.xml	D
3	Service	BroadcastReceiver	Activity	Intent	A
4	To represent a screen or user interface	To store and retrieve persistent data	To communicate between components and perform actions	To define the layout and appearance of an activity	C
5	READ_EXTERNAL_STORAGE	WRITE_EXTERNAL_STORAGE	CAMERA	INTERNET	C
6	To define the user interface layout of the app	To specify the permissions required by the app	To store and retrieve app data	To define the logic and behavior of the app	B
7	onPause()	onStop()	onDestroy()	onDetach()	B
8	LinearLayout	RelativeLayout	GridLayout	FrameLayout	C
9	To manage the dependencies and build process of the app	To design the user interface of the app	To handle version control of the app's source code	To test and debug the app	C
10	SharedPreferences	SQLite database	Internal storage	External storage	B

Exercise 2 : (10 Points)

Create a Login activity in Android Studio using Java and XML. The Login activity should consist of two input fields for the user to enter their username and password. Additionally, include a "Login" button that performs a simple validation of the username and password.

Requirements:

1. XML Layout: : (4.5 Points)

- * Create an XML layout file for the Login activity.
- * Include two EditText views for the username and password input fields.
- * Add a Button view for the "Login" button.

Listing 1: activity_login.xml

```
<!-- activity_login.xml -->

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

    <EditText
        android:id="@+id/editText_username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Username" />

    <EditText
        android:id="@+id/editText_password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:hint="Password" />

    <Button
        android:id="@+id/button_login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Login" />

</LinearLayout>
```

2. Java Code: : (5.5 Points)

- * Create a Java class for the Login activity.
- * Associate the XML layout with the Java class using setContentView() method.
- * Implement a click listener for the "Login" button.
- * Validate the username and password entered by the user.
- * Display a toast message indicating whether the login was successful or not.

Listing 2: LoginActivity.java

```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

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

```

public class LoginActivity extends AppCompatActivity {

    private EditText usernameEditText;
    private EditText passwordEditText;
    private Button loginButton;

    // Define the static username and password for validation
    private static final String VALID_USERNAME = "admin";
    private static final String VALID_PASSWORD = "password";

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

        // Bind the Java variables to the corresponding XML views using
        findViewById()
        usernameEditText = findViewById(R.id.editText_username);
        passwordEditText = findViewById(R.id.editText_password);
        loginButton = findViewById(R.id.button_login);

        loginButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String enteredUsername =
                    usernameEditText.getText().toString();
                String enteredPassword =
                    passwordEditText.getText().toString();

                // Perform basic validation
                if (!enteredUsername.isEmpty() &&
                    !enteredPassword.isEmpty()) {
                    // Compare the entered username and password with
                    the valid values
                    if (enteredUsername.equals(VALID_USERNAME) &&
                        enteredPassword.equals(VALID_PASSWORD)) {
                        // Display a toast message for successful login
                        Toast.makeText(LoginActivity.this, "Login_
                            Successful", Toast.LENGTH_SHORT).show();
                    } else {
                        // Display a toast message for failed login
                        Toast.makeText(LoginActivity.this, "Invalid_
                            Username_or_Password",
                            Toast.LENGTH_SHORT).show();
                    }
                } else {
                    // Display a toast message for empty fields
                    Toast.makeText(LoginActivity.this, "Both_fields_are_
                        required", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
}

```

3. Validation:

- * Perform basic validation to check if the username and password are not empty.
- * Define a static username and password in the Java class for comparison.
- * Compare the entered username and password with the static values.

* Display a toast message with appropriate feedback based on the validation result.

GOOD LUCK.
DR. MEFTAH ZOUAI