Lab Cycle: 03 Date:

EXPERIMENT NO: 08

AIM: Develop an application that uses ArrayAdapter with ListView.

Procedure:

MainActivity.java

```
package com.example.listviewapp;
import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.ListView;
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 ListView in the layout file by its ID.
        ListView listView = findViewById(R.id.listView);

        // This is the data we want to display in the ListView.
        String[] fruits = {"Apple", "Banana", "Cherry", "Date", "Grape", "Kiwi", "Lemon", "Mango",
"Orange", "Peach"};

        // Create an ArrayAdapter to bind the data (the 'fruits' array) to the ListView.
        // The ArrayAdapter needs three things:
        // 1. The current context (this)
        // 2. The layout for a single list item (we use a simple built-in Android layout)
        // 3. The data array to display
```

```
                ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
                android.R.layout.simple_list_item_1, fruits);

        // Set the adapter on the ListView. This is the crucial step that populates the list.
        listView.setAdapter(adapter);
    }
}
```

activity_main.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <!-- This is the ListView component. -->
    <!-- It is the container that will display the list of items. -->
    <!-- The width and height are set to 'match_parent' to fill the entire screen. -->
    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:

RESULT:

The program was executed successfully and the output was verified.

Lab Cycle: 03 Date:

EXPERIMENT NO: 09

AIM: Implement Options Menu to navigate to activities.

Procedure:

MainActivity.java

```
package com.example.optionsmenuapp;

import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

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

    // This method is responsible for creating the options menu.
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu XML file into the Menu object.
        getMenuInflater().inflate(R.menu.menu_main, menu);
```

```
        return true;
    }

    // This method is called every time a menu item is selected.
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        // Get the ID of the selected menu item.
        int id = item.getItemId();

        // Use a conditional statement to determine which activity to launch.
        if (id == R.id.action_activity_one) {
            // Create an Intent to start ActivityOne.
            Intent intent = new Intent(MainActivity.this, ActivityOne.class);
            startActivity(intent);
            Toast.makeText(this, "Navigating to Activity One", Toast.LENGTH_SHORT).show();
            return true;
        } else if (id == R.id.action_activity_two) {
            // Create an Intent to start ActivityTwo.
            Intent intent = new Intent(MainActivity.this, ActivityTwo.class);
            startActivity(intent);
            Toast.makeText(this, "Navigating to Activity Two", Toast.LENGTH_SHORT).show();
            return true;
        }

        // Pass the selection to the superclass for default handling if no matches are found.
        return super.onOptionsItemSelected(item);
    }
}
```

activity_main.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Main Activity"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

activityOne.java

```
package com.example.optionsmenuapp;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

// This is the first destination activity.
public class ActivityOne extends AppCompatActivity {
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_one); // Display the layout for this activity.
    }
}
```

activityOne.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ActivityOne">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is Activity One"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

activityTwo.java

```
package com.example.optionsmenuapp;

import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

// This is the second destination activity.
public class ActivityTwo extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_two); // Display the layout for this activity.
    }
}
```

activityTwo.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".ActivityTwo">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is Activity Two"
        android:textSize="24sp"
```

```
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

menu_main.xml

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

    <!-- This item will navigate to the first activity. -->
    <item
        android:id="@+id/action_activity_one"
        android:title="Go to Activity One"
        app:showAsAction="never" />

    <!-- This item will navigate to the second activity. -->
    <item
        android:id="@+id/action_activity_two"
        android:title="Go to Activity Two"
        app:showAsAction="never" />
</menu>
```

OUTPUT:

RESULT:

The program was executed successfully and the output was verified.

Lab Cycle: 03 Date:

EXPERIMENT NO: 10

AIM: Develop application that works with explicit intents

Procedure:

MainActivity.java

```
package com.example.explicitintentapp;

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 {
    private Button navigateButton;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main); // Set the layout for the main activity

        // Find the button from the layout file by its ID.
        navigateButton = findViewById(R.id.navigateButton);
        // Set an OnClickListener on the button to handle clicks.
        navigateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // An explicit intent is used to launch a specific component (Activity).
                // It requires two arguments: the current context and the target class.
                Intent explicitIntent = new Intent(MainActivity.this, SecondActivity.class);
```

```
            // Use startActivity() to execute the intent and launch the new activity.
            startActivity(explicitIntent);
        }
    });
  }
}
```

activity_main.xml

```
<?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:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
  <!-- The button that, when clicked, will trigger the explicit intent. -->
  <Button
    android:id="@+id/navigateButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Go to Second Activity"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

SecondActivity.java

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

```
// SecondActivity is the destination of our explicit intent.
public class SecondActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_second); // Set the layout for this activity.
    }
}
```

SecondActivity.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".SecondActivity">

    <!-- A simple TextView to confirm we've successfully navigated to this activity. -->
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Welcome to the Second Activity!"
        android:textSize="24sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

OUTPUT:

RESULT:

The program was executed successfully and the output was verified.

Lab Cycle: 03 Date:

EXPERIMENT NO: 11

AIM: Develop an application that implements Spinner component and perform event handling

Procedure:

MainActivity.java

```
package com.example.spinnerapp;


import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;
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 from activity_main.xml
      // Find the Spinner from the layout file by its ID.
      Spinner spinner = findViewById(R.id.languageSpinner);
      // Create an ArrayAdapter using the string array from strings.xml and a default spinner layout.
      ArrayAdapter<CharSequence> adapter = ArrayAdapter.createFromResource(
            this,
            R.array.languages_array, // Reference to the string array defined in strings.xml
            android.R.layout.simple_spinner_item // A default layout for the spinner's items
      );
      // Specify the layout to use when the list of choices appears.
      adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
      // Apply the adapter to the spinner.
      spinner.setAdapter(adapter);
```

```
        // Set an event listener to handle item selection.
        spinner.setOnItemSelectedListener(new AdapterView.OnItemSelectedListener() {
            @Override
            public void onItemSelected(AdapterView<?> parent, View view, int position, long id) {
                // Get the selected item's text from the parent AdapterView.
                String selectedItem = parent.getItemAtPosition(position).toString();
                // Show a toast message with the selected item.
                Toast.makeText(MainActivity.this, "Selected: " + selectedItem,
Toast.LENGTH_SHORT).show();
            }

            @Override
            public void onNothingSelected(AdapterView<?> parent) {
                // This method is called when no item is selected.
                // You can add logic here if needed.
            }
        });
    }
}
```

activity_main.xml

```
<?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:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:padding="16dp">
```

```
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Select a programming language:"
        android:textSize="18sp"
        app:layout_constraintBottom_toTopOf="@+id/languageSpinner"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_chainStyle="packed"/>

    <!-- This is the Spinner component. Its items will be populated via an ArrayAdapter. -->
    <Spinner
        android:id="@+id/languageSpinner"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Spinner App</string>
    <!-- This is the string array that will be used to populate the Spinner. -->
    <string-array name="languages_array">
```

```
        <item>Java</item>
        <item>Python</item>
        <item>C++</item>
        <item>Kotlin</item>
        <item>Swift</item>
        <item>JavaScript</item>
        <item>Dart</item>
    </string-array>
</resources>
```

OUTPUT:

RESULT:

The program was executed successfully and the output was verified.