



---

## CP 313: MOBILE APPLICATIONS DEVELOPMENT

# Contents

---

## ○Activities

- Defining an activity
- Starting a new activity with an intent
- Passing data between activities with extras
- Navigating between activities

# What is an Activity?

---

- An Activity is an application component
  - Represents one window, one hierarchy of views
  - Typically fills the screen, but can be embedded in other activity or appear as floating window
  - Java class, typically one activity in one file

# What is an Activity?

---

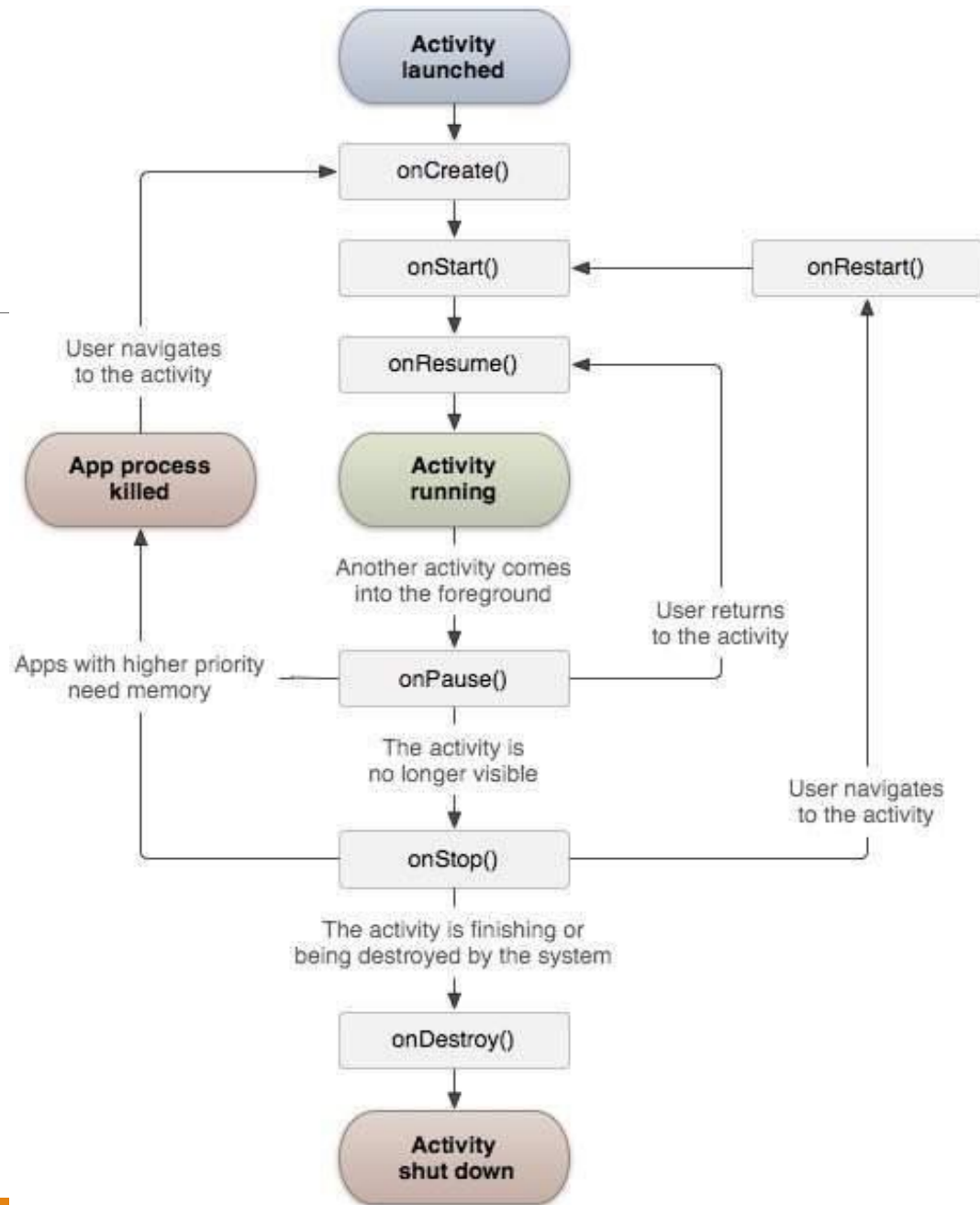
- An activity represents a single screen with a user interface just like window or frame of Java.
- Android activity is the subclass of `ContextThemeWrapper` class.

# Activity Life Cycle

---

- If you have worked with C, C++ or Java programming language then you must have seen that your program starts from main() function.
- Very similar way, Android system initiates its program with in an Activity starting with a call on onCreate() callback method.
- There is a sequence of callback methods that start up an activity and a sequence of callback methods that tear down an activity as shown in the next slide.

# Activity Life Cycle



# Activity Life Cycle

---

- ❑ **onCreate():** This is the first callback and called when the activity is first created.
- ❑ **onStart():** This callback is called when the activity becomes visible to the user.
- ❑ **onResume():** This is called when the user starts interacting with the application.
- ❑ **onPause():** The paused activity does not receive user input and cannot execute any code and called when the current activity is being paused and the previous activity is being resumed.
- ❑ **onStop():** This callback is called when the activity is no longer visible.
- ❑ **onDestroy():** This callback is called before the activity is destroyed by the system.
- ❑ **onRestart():** This callback is called when the activity restarts after stopping it.

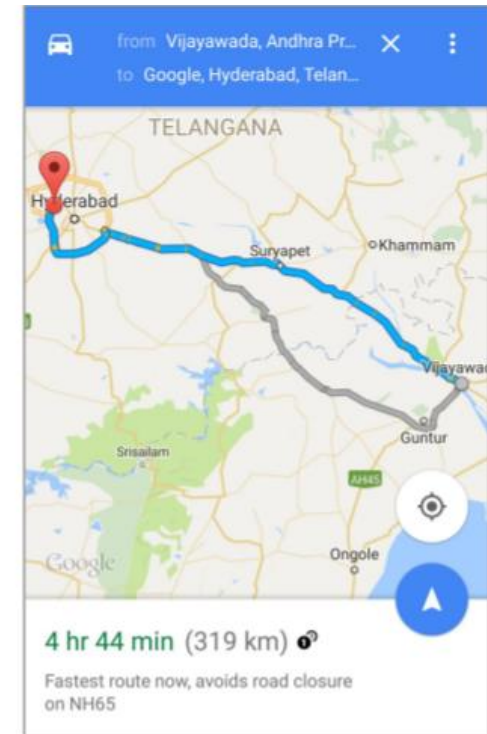
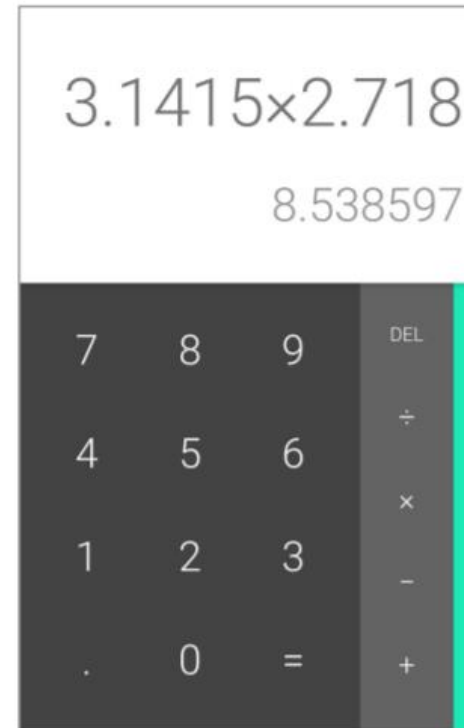
# What does an Activity do?

---

- Represents an activity, such as ordering groceries, sending email, or getting directions
  - Handles user interactions, such as button clicks, text entry, or login verification
  - Can start other activities in the same or other apps
  - Has a life cycle—is created, started, runs, is paused, resumed, stopped, and destroyed



# Examples of Activities



# Apps and activities

---

- Activities are loosely tied together to make up an app
  - First activity user sees is typically called "main activity"
  - Activities can be organized in parent-child relationships in the Android manifest to aid navigation

# Layouts and Activities

---

- An activity typically has a UI layout
  - Layout is usually defined in one or more XML files
  - Activity "inflates" layout as part of being created

# Implement new activities

---

1. Define layout in XML
2. Define Activity Java class
  - extends AppCompatActivity
3. Connect Activity with Layout
  - Set content view in onCreate()
4. Declare Activity in the Android manifest

# 1. Define layout in XML

---

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
    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:id="@+id/txtviewFName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="First Name"
        android:layout_marginLeft="20dp"
        android:textSize="40dp"
        android:layout_marginTop="20dp"
        android:textStyle="bold"
        android:textColor="@color/purple_700">
    </TextView>
```

## 2. Define Activity Java class

---

```
package com.example.myapplication10;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.os.Bundle;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

### 3. Connect activity with layout

---

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```

Resource is layout in this XML file



### 3. Connect activity with layout

---

```
package com.example.myapplication10;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
  
import android.os.Bundle;  
  
import android.view.View;  
  
import android.widget.Button;  
  
import android.widget.EditText;
```



# 3. Connect activity with layout

---

```
public class MainActivity extends AppCompatActivity {  
    private EditText FirstName, LastName, Results;  
    private Button Submit;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        FirstName = (EditText) findViewById(R.id.editTextFName);  
        LastName = (EditText) findViewById(R.id.editTextLName);  
        Results = (EditText) findViewById(R.id.tvResults);  
        Submit = (Button) findViewById(R.id.btnSubmit);  
    }  
}
```

# 3. Connect activity with layout

---

```
Submit.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        int num1=Integer.parseInt(FirstName.getText().toString());  
        int num2=Integer.parseInt(LastName.getText().toString());  
        int sum=num1+num2;  
        Results.setText("The Answer is "+sum);  
    }  
});  
  
}  
}
```

## 4. Declare activity in Android manifest

---

```
<activity android:name=".MainActivity">
```

## 4. Declare main activity in manifest

---

```
android:theme="@style/Theme.MyApplication" />
<activity
    android:name=".MainActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

---

# Intents



# What is an intent?

---

- An intent is a description of an operation to be performed.
- An Intent is an object used to request an action from another app component via the Android system.

# What can intents do?

---

- Start activities

- A button click starts a new activity for text entry
- Clicking Share opens an app that allows you to post a photo

- Start services

- Initiate downloading a file in the background

- Deliver broadcasts

- The system informs everybody that the phone is now charging



# Types of Intents

---

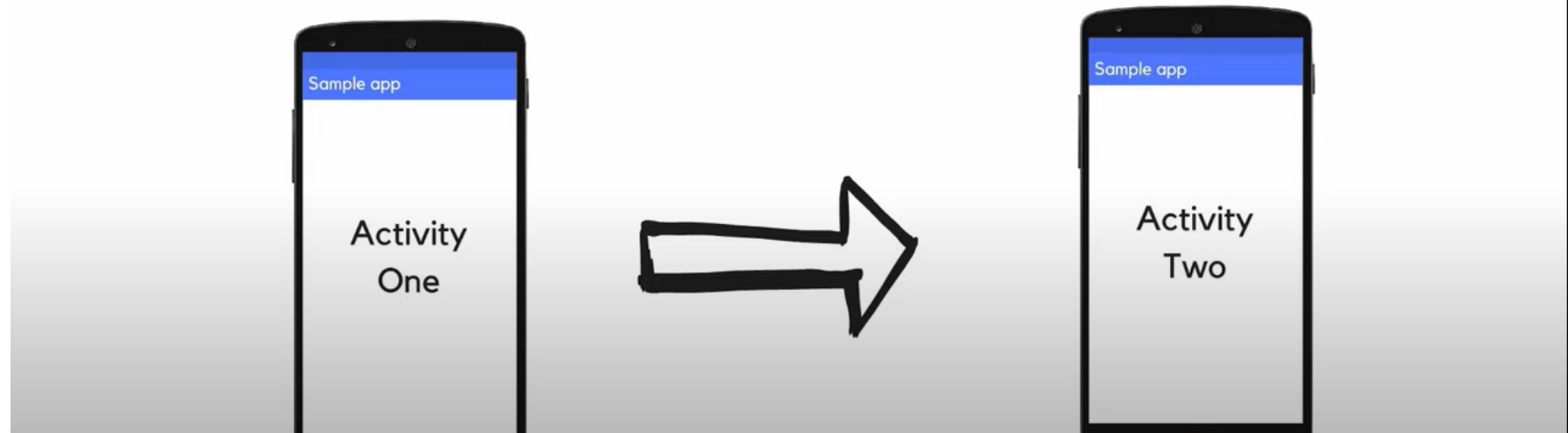
- Explicit Intent

- Implicit Intent

# Types of Intents (Explicit Intent)

---

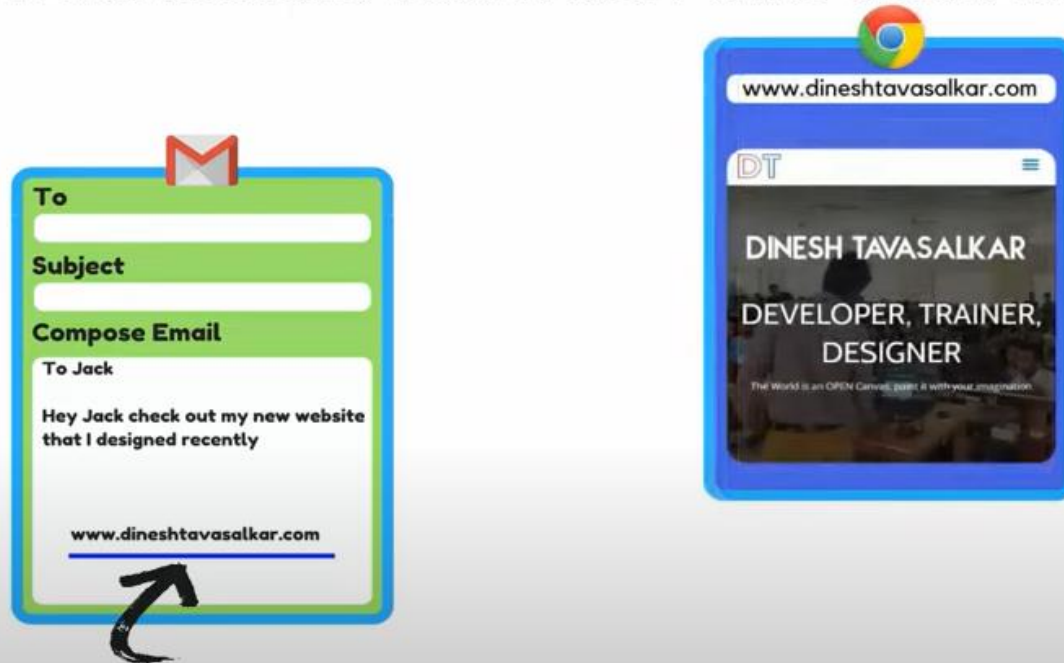
communicates between two activities inside  
the same application



We can also pass the information from one activity to another using explicit intent.

# Types of Intents (Implicit Intent)

Communicates between two activities of different application



---

# Starting Activities



# Start an Activity with an explicit intent

---

To start a specific activity, use an explicit intent

## 1. Create an intent

- `Intent intent = new Intent(this, ActivityName.class);`

## 2. Use the intent to start the activity

- `startActivity(intent);`

```
public void onClick(View v) {  
    }  
});  
  
}  
public void Activity2(){  
    Intent myIntent = new Intent(this, HomeActivity.class);  
    startActivity(myIntent);  
}  
}
```

# Sending and Receiving Data

---

- ❑ We can send data while calling one activity from another activity using intent.
- ❑ All we have to do is add the data to Intent object using `putExtra()` method.
- ❑ The data is passed in `key value pair`.
- ❑ The value can be of types like `int, float, long, string, etc.`

# Sending Data

```
Submit.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
  
        Activity2();  
  
    }  
});  
}  
public void Activity2(){  
    int num1=Integer.parseInt(FirstName.getText().toString());  
    int num2=Integer.parseInt(LastName.getText().toString());  
    //int sum=num1+num2;  
    // Results.setText("The Answer is "+sum);  
    Intent myIntent = new Intent(this,HomeActivity.class);  
    myIntent.putExtra("numberOne:",num1);  
    myIntent.putExtra("numberTwo:",num2);  
    startActivity(myIntent);  
}  
}
```



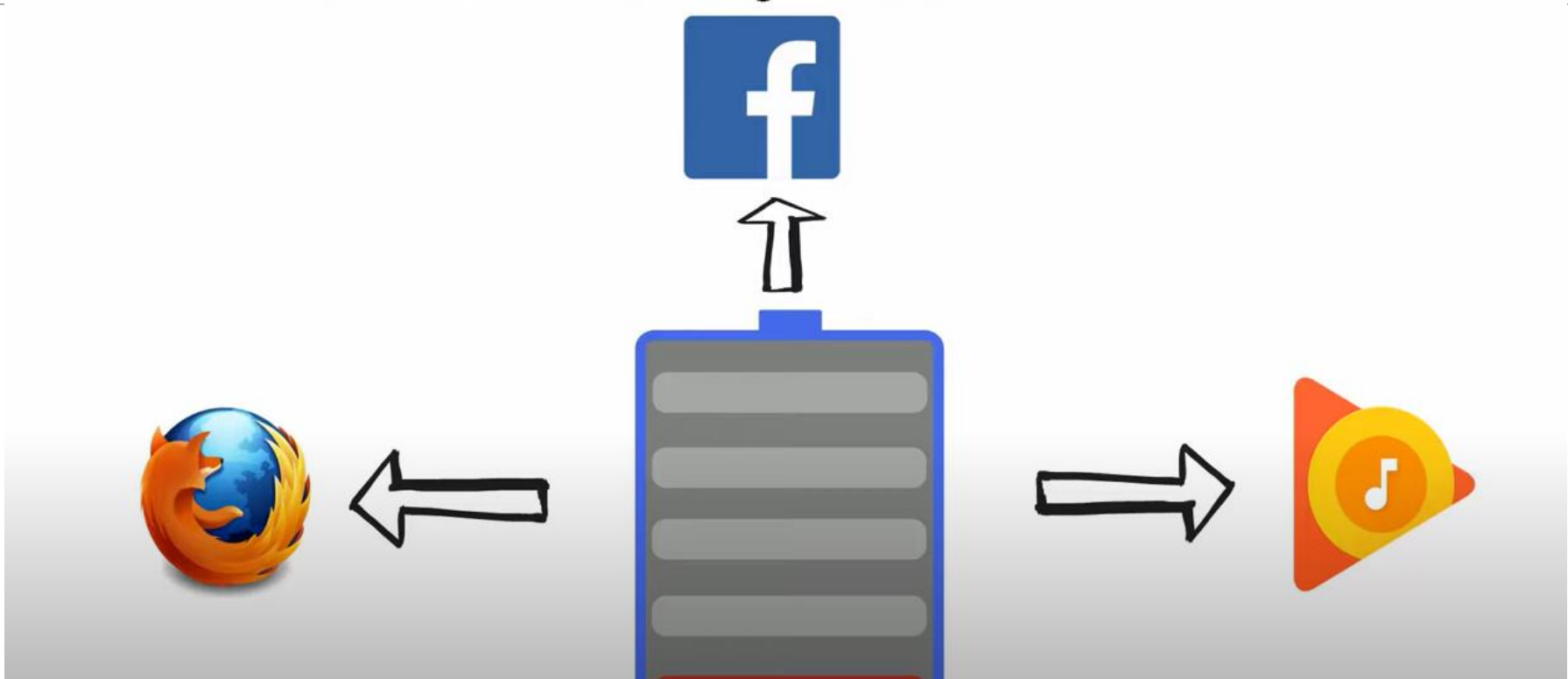
# Receiving Data

```
public class HomeActivity extends AppCompatActivity {  
    private TextView n1,n2;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_home);  
        n1=(TextView)findViewById(R.id.tvNumber1);  
        n2=(TextView)findViewById(R.id.tvNumber2);  
  
        Intent intent =getIntent();  
        int number_one = Integer.parseInt(intent.getStringExtra("numberOne"));  
        int number_two = Integer.parseInt(intent.getStringExtra("numberTwo"));  
        n1.setText(number_one);  
        n2.setText(number_two);  
  
    }  
}
```

# Intents can be used for broadcasting a message

---

- Suppose that the battery of your mobile phone is low and the message is sent to different android applications to your phone alerting that the battery is low



# Your Turn

---

- Services
- Broadcast Message
- Implicit Intents

---

END

