**PRACTICAL # 07**

**OBJECT:**

Understanding the Intents

**THEORY:**

We have been working with intents in Lab 5, but without discussing its purpose. In this lab we will cover the concepts and functionality of intents. As the name implies, it’s an intention to do an action.

It’s a message to say: I want something to happen

When you create an intent, you are saying that you want to switch from one activity to another.

An intent is an object used to communicate with the android OS. Intents are used with activities, services, broadcast receivers, and content providers. You can use intents to tell the Android OS which activity to start.

Intent Types:

**Explicit intents** are used with a Context and Class object to start an activity within your application.

**Implicit intents** are used to start activities outside of your application.

**Explicit Intent (Constructor):**

An intent class has many constructors (for starting a new activity within the application), but a typical constructor has following two arguments:

The first argument contains the current activity.

The second argument contains the class name of the activity to be started.

The following intent starts the second activity (Main2Activity.class) from MainActivity.this



**Intent-filter:**

Specifies the types of intents that an activity, service, or broadcast receiver can respond to.

An intent filter declares the capabilities of its parent component (what an activity or service can do and what types of broadcasts a receiver can handle). It opens the component to receiving intents of the advertised type, while filtering out those that are not meaningful for the component. Most of the contents of the filter are described by its <action>, <category>, and <data> sub-elements.

An example intent filter is system to “Start the Send Email activity in the Gmail app":

*<activity android:name=".ExampleActivity" android:icon="@drawable/app\_icon"> <intent-filter>*

*<action android:name="android.intent.action.SEND" />*

*<data android:mimeType="text/plain" />*

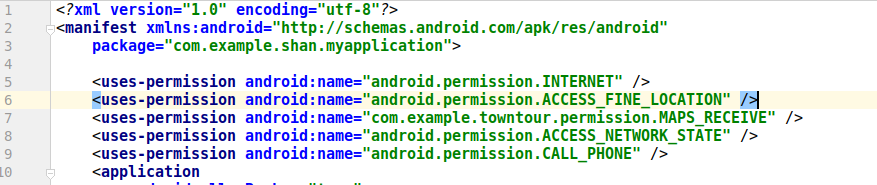
*</intent-filter>*

*<activity>*

**Permissions:**

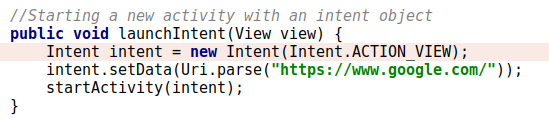
Permits an app or its components to use system resources such as Camera, Internet etc. Manifest file contains these permissions. A permission is declared as <uses-permission>, and it can be either declared globally or for a particular activity.

In case of local permission of an activity, the calling activity must have a matching <uses-permission> element. Some of the permissions that the application requires are shown below.

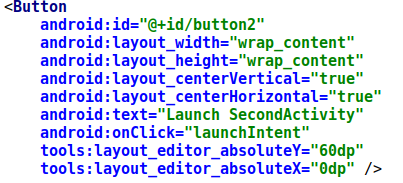


**Implicit intents:**

To launch an external activity, use intent with appropriate class (eg. ACTION\_VIEW):

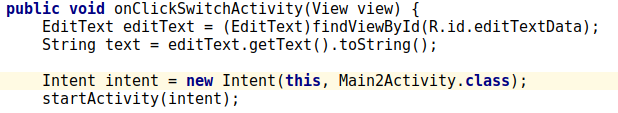


Use the following code in XML to create a button element and add the onClick callback:



To use an explicit intent, startActivity is called passing an Intent object.

The ActivityManager uses that intent to determine the activity to start.



**ACTIVITIES**

**Activity 1**

Create a simple program that takes input a URL in a TextView and starts an activity with explicit intent to browse that URL.

**REVIEW QUESTIONS**

1. What are intents?
2. What are intent-filters?
3. What is the difference between explicit and implicit intent?
4. Why do we add permissions in the manifest file?