

Android Intents and Software Version Control

EE5415

Mobile Apps Design and Development

Canvas Quiz on Week 6

- A Canvas Quiz containing 25 multiple choice questions will be released on the next Wednesday (March 4, 2020) at 3:00 pm.
- As this quiz will be closed, students must complete this quiz before 11:00 pm, tonight.

Agenda

- Android Intents
- Android Permission
- Lab04 : Intents and Activity Lifecycle
- Software Version Control: Git and GitHub
- **Canvas Quiz (25 questions)**
 - Released at 3:00pm
 - Students have to complete this quiz on-line before 11:00pm as the quiz will be closed.

Android Intents

- Intent Demo App
- Starting activities with Intents
- Explicit Activation
- Implicit Activation
- Intent Resolution
- Intent Filter

The Intent Class

- A data structure that represents
 - An operation to be performed, or
 - An event that has occurred
- We only focus on using Intents to specify operations to be performed, not for event notification
 - i.e. Intents used to start a single activity

Intents as Desired Operations

- Intents provide a flexible language for specifying operations to be performed
 - e.g. Pick a contact, take a photo, dial a phone number
- Intent is constructed by one component that wants some work done
- Received by one activity that can perform that work

Intent Fields

- Component
- Action
- Data
- Category
- Type
- Extras
- Flags

Component

- The component that should receive this intent
- Use this when there's exactly one component that should receive the intent
- Setting the Component
 - Intent myIntent = new Intent(Context packageContext, Class<?>cls);
 - Example in BMI project

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

or

 - Intent myIntent = new Intent ();
and one of:
myIntent.setComponent(), setClass(), or setClassName()

Action

- String representing desired operation
- Examples:
 - ACTION_CALL : Call a phone number
 - ACTION_VIEW : Display a View such as webpage
 - ACTION_SYNC : Synchronize device data with server
 - ACTION_MAIN : Start as initial activity of app
- Setting the Intent Action
 - Intent myIntent = new Intent(Intent.ACTION_CALL);
OR
 - Intent myIntent = new Intent;
▪ myIntent.setAction(Intent.ACTION_CALL);

Data

- Data associated with the Intent
 - Formatted as a Uniform Resource Identifier (URI)
- Examples
 - A webpage address to be view on a browser
 - Uri.parse("http://developer.android.com") ;
 - Number to dial in the phone dialer
 - Uri.parse("tel:34426789")
- Setting Intent data
 - Intent myIntent = new Intent(Intent.ACTION_CALL,Uri.parse("tel:34426789")) ;
OR
 - Intent myIntent = new Intent(Intent.ACTION_CALL) ;
 - myIntent.setData(Uri.parse("tel:34426789")) ;

Category

- Additional information about the components that can handle the intent
- Examples
 - Category_browsable
 - can be invoked by a browser to display data ref's by a URI
 - Category_launcher
 - can be the initial activity of a task & is listed in top-level app launcher

Type

- Specifies the MIME type of the Intent data
- Examples
 - image/*, image/png, image/jpeg,
 - text/html, text/plain
 - If unspecified, Android will infer the type
- Setting the Type
 - `Intent.setType(String type)`
 - or
 - `Intent.setDataAndType(Uri data, String type)`

Extras

- Additional information associated with Intent
Treated as a map (key-value pairs)
- Examples
 - Intent.EXTRA_EMAIL: email recipients

```
Intent myIntent = new Intent(Intent.ACTION_SEND);
myIntent.putExtra(android.content.Intent.EXTRA_EMAIL,
new String []{
    "peter@gmail.com", "ryan@microsoft.com",
    "mary@.gov.hk", "emily@yahoo.com"
}
);
```

Setting the Extra Attribute

- Several forms depending on data type
- `putExtra(String name, String value);`
- `putExtra(String name, float[] value);`
- . . .
- Examples in BMI App using Bundle:

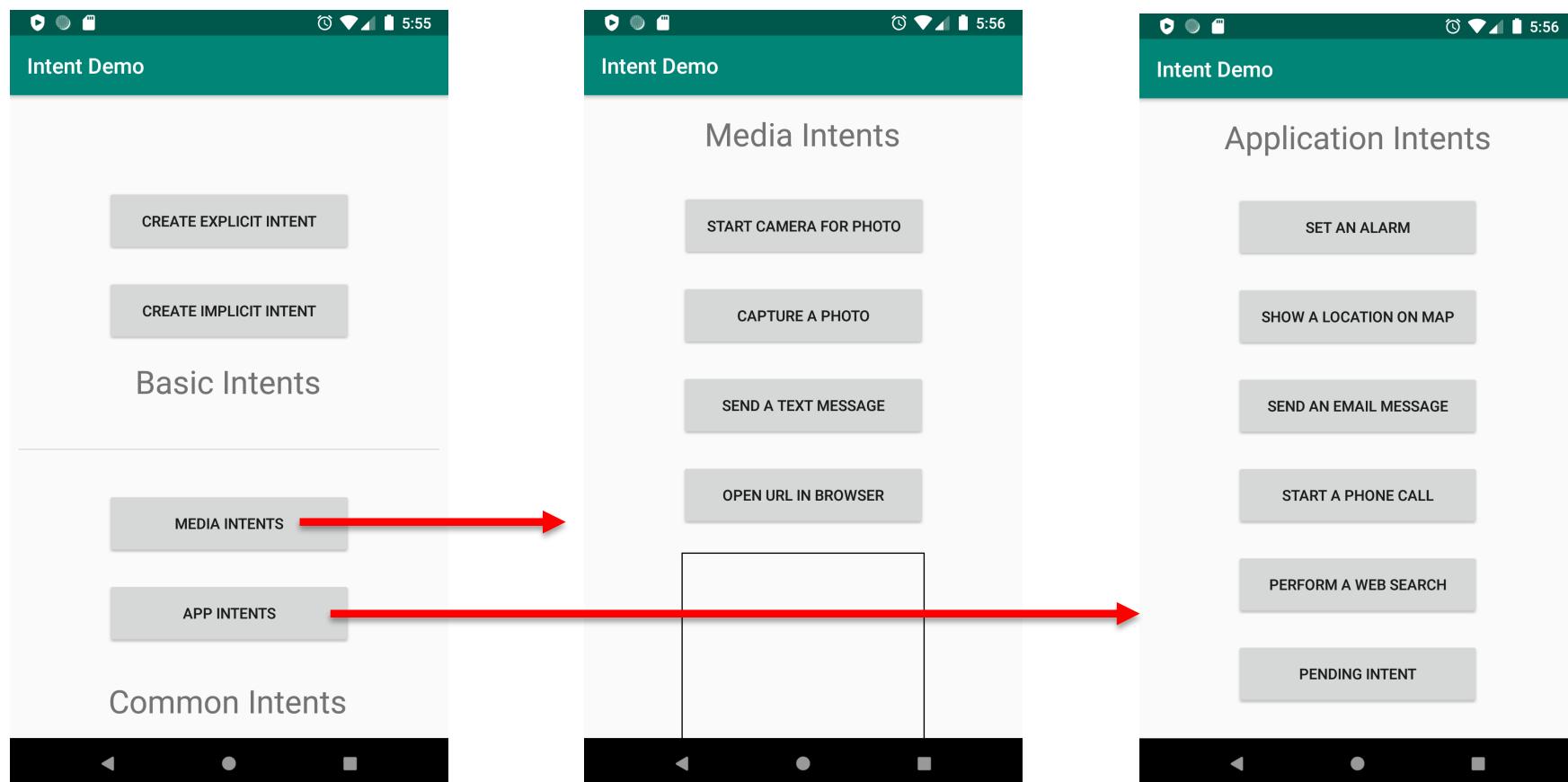
```
Intent intent = new Intent(this, ReportActivity.class);
        Bundle bundle = new Bundle();
        bundle.putString("height", height);
        bundle.putString("weight", weight);
        intent.putExtras(bundle);
```

Flags

- Specify how Intent should be handled
- Examples
 - FLAG_ACTIVITY_NO_HISTORY
 - Don't put this Activity in the History stack
 - FLAG_DEBUG_LOG_RESOLUTION
 - Print extra logging information when this Intent is processed
- Setting Flags
 - Intent myIntent = new Intent(Intent.ACTION_SEND);
 - myIntent.setFlags(Intent.FLAG_ACTIVITY_NO_HISTORY);

Intent Demo App

- Source Code is available in the course website
- Run it on AVD with API 27 or below:

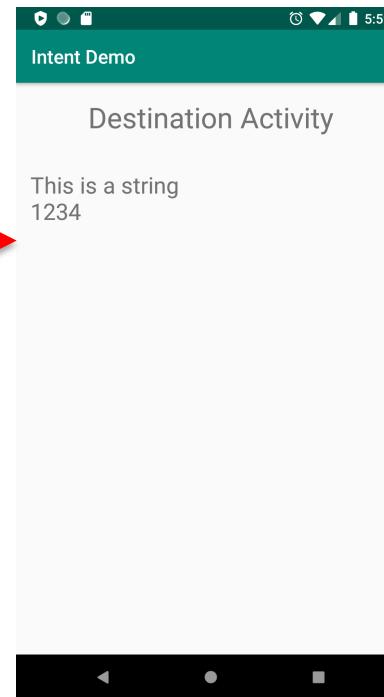
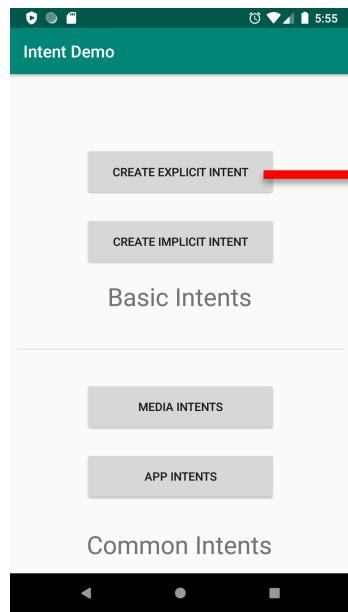


Starting Activities with Intents

- `startActivity(Intent intent, ...)`
- `startActivityForResult(Intent intent, ...)`
- The Target Activity
 - Can be named explicitly by setting the intent's component
 - Can be determined implicitly

Explicit Intent

```
if (viewClicked == R.id.create_explicit) {  
    // Build an explicit Intent to launch our Activity  
    Intent i = new Intent(packageContext: this, DestinationActivity.class);  
  
    // send data along with the Intent to the destination  
    i.putExtra(name: "IntData", value: 1234);  
    i.putExtra(name: "StringData", value: "This is a string");  
  
    // Start the activity with our explicit intent  
    startActivity(i);  
}
```



Implicit Activation

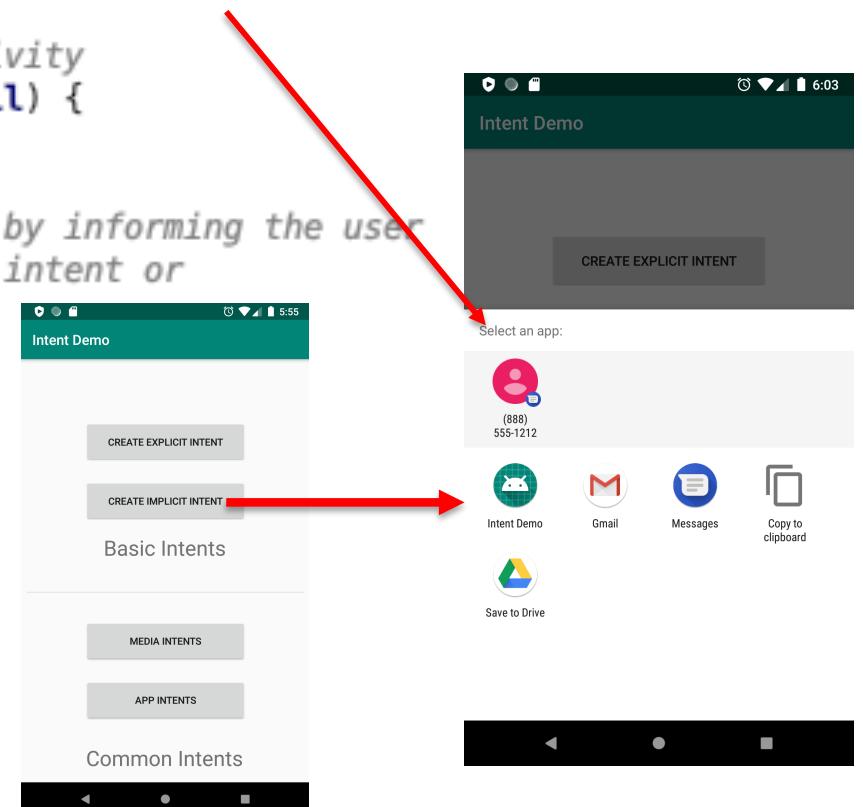
- When the activity to be activated is not explicitly named, Android tries to find activities that match the Intent
- This process is called Intent Resolution

Implicit Intent

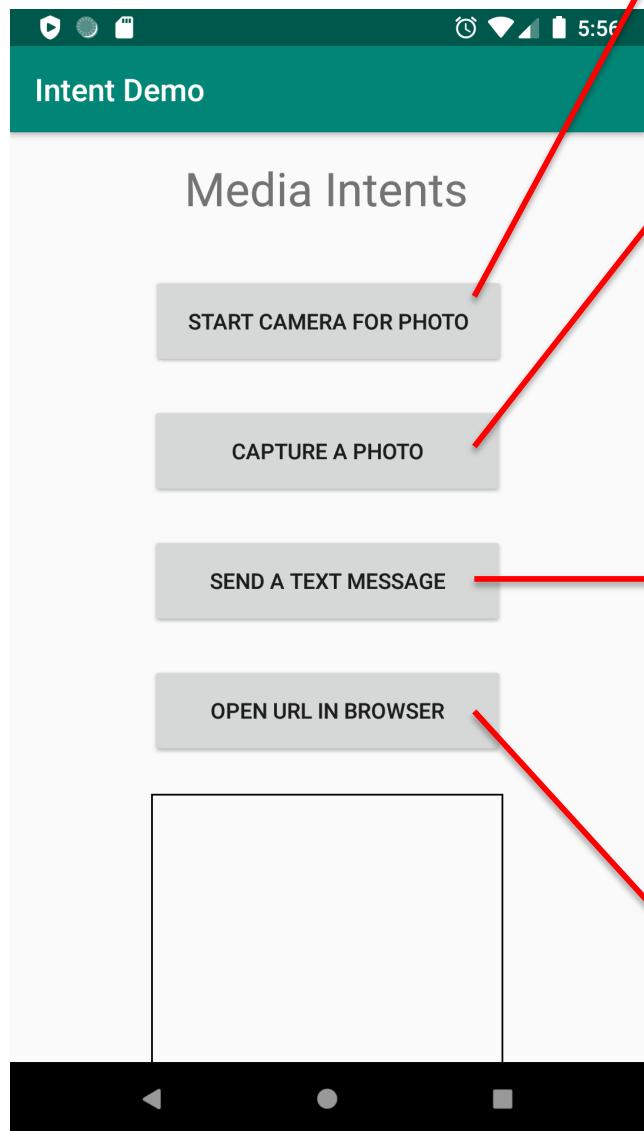
```
else if (viewClicked == R.id.create_implicit) {
    // Build an implicit intent to handle a type of action
    String textMessage = "This is a sample message";
    Intent i = new Intent();
    i.setAction(Intent.ACTION_SEND);
    i.setType("text/plain");
    i.putExtra(Intent.EXTRA_TEXT, textMessage);

    // use an intent chooser to force a choose dialog
    Intent chooser = Intent.createChooser(i, title: "Select an app:");

    // Verify that the intent will resolve to an activity
    if (i.resolveActivity(getApplicationContext()) != null) {
        startActivity(chooser);
    }
    // Typically you would handle the null case here by informing the user
    // that there is no installed app to handle this intent or
    // by taking some other action
}
```



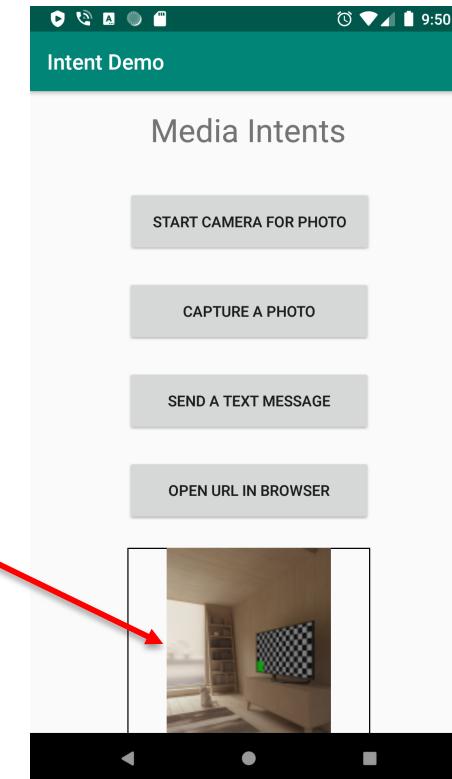
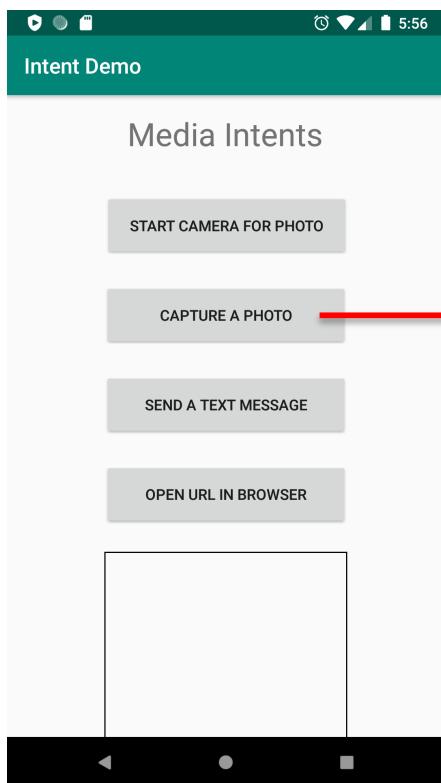
Media Intents



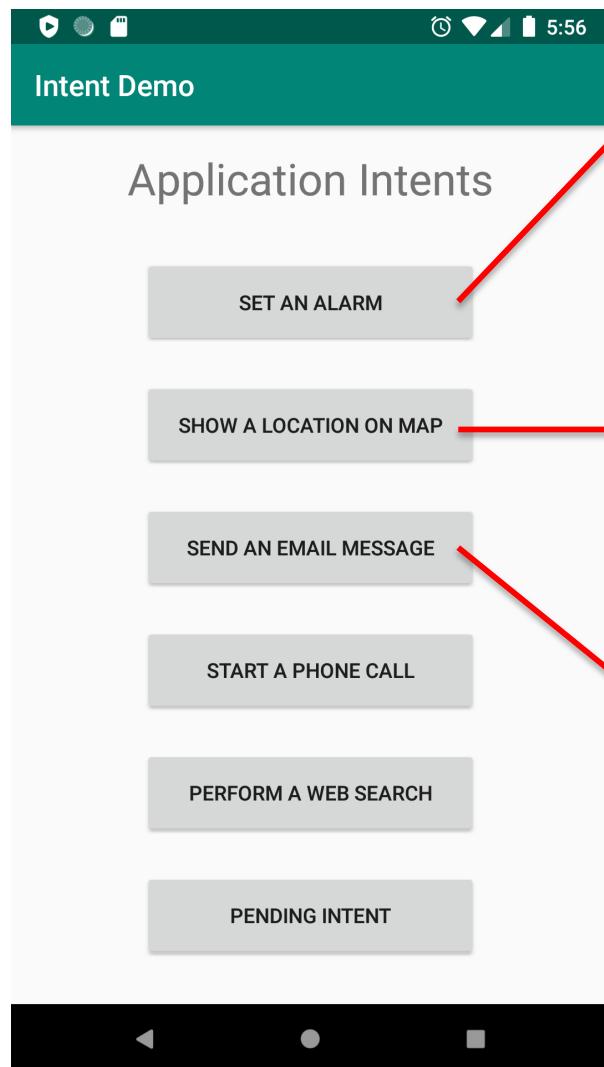
```
if (btnClick == R.id.btnStartCamera) {  
    // TODO: Start the camera in photo mode  
    Intent i = new Intent(MediaStore.INTENT_ACTION_STILL_IMAGE_CAMERA);  
  
    if (i.resolveActivity(getApplicationContext()) != null) {  
        startActivityForResult(i);  
    }  
}  
else if (btnClick == R.id.btnCapturePic) {  
    // Take a picture and consume the returned result bitmap  
    Intent i = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);  
  
    if (i.resolveActivity(getApplicationContext()) != null) {  
        // start the activity and indicate that we expect a result back  
        startActivityForResult(i, GET_IMAGE_CAPTURE);  
    }  
}  
else if (btnClick == R.id.btnSendText) {  
    final String message = "This is a text message";  
  
    Intent i = new Intent(Intent.ACTION_SENDTO);  
  
    // Use the setData function to indicate the type of data that will be sent  
    // this will help the system figure out what apps to include in the chooser  
    i.setData(Uri.parse("sms:18885551212"));  
    i.putExtra("name", "sms_body", message);  
  
    if (i.resolveActivity(getApplicationContext()) != null) {  
        startActivityForResult(i);  
    }  
}  
else if (btnClick == R.id.btnOpenURL) {  
    String url = "http://www.google.com";  
  
    // Parse the URL string using the Uri class  
    Uri webpage = Uri.parse(url);  
  
    Intent i = new Intent(Intent.ACTION_VIEW, webpage);  
  
    if (i.resolveActivity(getApplicationContext()) != null) {  
        startActivityForResult(i);  
    }  
}
```

Capture Photo

```
// This function will be called when an activity that was started for the purpose  
// of returning a result has some data for our app to process  
  
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    if (requestCode == GET_IMAGE_CAPTURE && resultCode == RESULT_OK) {  
        // Retrieve the data from the result intent and look for the bitmap  
        Bundle extras = data.getExtras();  
        Bitmap imageBitmap = (Bitmap) extras.get("data");  
        imgView.setImageBitmap(imageBitmap);  
    }  
}
```

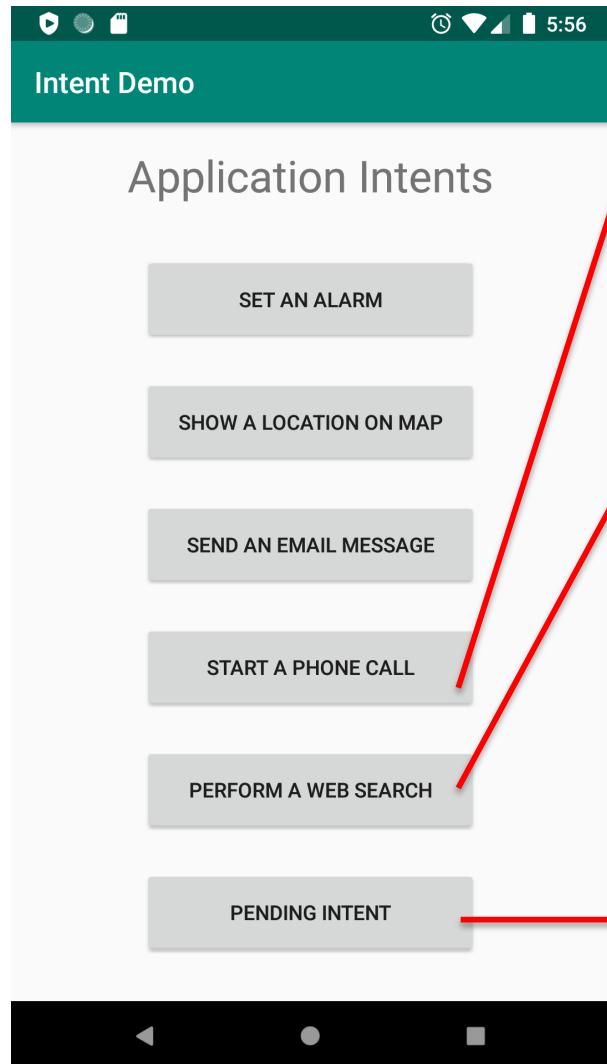


Apps Intents (1)



```
if (btnClick == R.id.btnSetAlarm) {  
    final String message = "Time to wake up!";  
    final int hour = 6;  
    final int minutes = 30;  
  
    // Create an intent to tell the system to set an alarm  
    // NOTE: your app needs to have the Set Alarm permission  
    Intent i = new Intent(AlarmClock.ACTION_SET_ALARM)  
        .putExtra(AlarmClock.EXTRA_MESSAGE, message)  
        .putExtra(AlarmClock.EXTRA_HOUR, hour)  
        .putExtra(AlarmClock.EXTRA_MINUTES, minutes)  
        .putExtra(AlarmClock.EXTRA_VIBRATE, true);  
  
    if (i.resolveActivity(getApplicationContext()) != null) {  
        startActivity(i);  
    }  
}  
  
else if (btnClick == R.id.btnShowMapLoc) {  
    // Locations can be specified using latlongs, queries, addresses, etc.  
    String location = "geo:37.4220,-122.0841";  
    String location = "geo:0,0?q=37.4220,-122.0841(GooglePlex)";  
    String location = "geo:0,0?q=20+W+34th+St+10001";  
    String location = "geo:47.6205,-122.3493?q=restaurants";  
  
    // Parse the location using the Uri class  
    Uri geoLocUri = Uri.parse(location);  
  
    // Pass the Uri directly to the Intent constructor  
    Intent i = new Intent(Intent.ACTION_VIEW, geoLocUri);  
  
    if (i.resolveActivity(getApplicationContext()) != null) {  
        startActivity(i);  
    }  
}  
  
else if (btnClick == R.id.btnSendAnEmail) {  
    String[] addresses = { "test@example.com" };  
    String[] ccs = { "someone@example.com" };  
    String subject = "This is a test";  
    String message = "This is a test email message!";  
  
    Intent i = new Intent(Intent.ACTION_SENDTO);  
  
    // Use setData to ensure that only email apps respond  
    i.setData(Uri.parse("mailto:"));  
  
    i.putExtra(Intent.EXTRA_EMAIL, addresses);  
    i.putExtra(Intent.EXTRA_SUBJECT, subject);  
    i.putExtra(Intent.EXTRA_CC, ccs);  
    i.putExtra(Intent.EXTRA_TEXT, message);  
  
    if (i.resolveActivity(getApplicationContext()) != null) {  
        startActivity(i);  
    }  
}
```

Apps Intents (2)



```
else if (btnClick == R.id.btnStartPhoneCall) {
    String phoneNumber = "1-800-555-1212";

    // Build the Uri for the phone number
    Uri numUri = Uri.parse("tel:" + phoneNumber);

    // Your application needs the CALL_PHONE permission for this intent
    Intent i = new Intent(Intent.ACTION_DIAL);
    Intent i = new Intent(Intent.ACTION_CALL); // Actually makes the call
    // Set the Uri as the intent data
    i.setData(numUri);

    if (i.resolveActivity(getApplicationContext()) != null) {
        startActivity(i);
    }
}

else if (btnClick == R.id.btnWebSearch) {
    String queryStr = "Eiffel Tower";

    // Create an intent to fire off a web search
    Intent i = new Intent(Intent.ACTION_WEB_SEARCH);

    i.putExtra(SearchManager.QUERY, queryStr);
    if (i.resolveActivity(getApplicationContext()) != null) {
        startActivity(i);
    }
}

else if (btnClick == R.id.btnPendingIntent) {
    NotificationCompat.Builder builder =
        new NotificationCompat.Builder( context: this, channelId: "my_channel");

    // Create the intent that will be fired when the user taps the notification
    Intent intent = new Intent( packageContext: this, DestinationActivity.class);
    intent.putExtra( name: "IntData", value: 5415);
    intent.putExtra( name: "StringData", value: "This comes from a notification.");
    // Wrap it up in a PendingIntent
    PendingIntent pendingIntent = PendingIntent.getActivity( context: this, NOTIFY_ID, intent
        PendingIntent.FLAG_CANCEL_CURRENT);

    builder.setSmallIcon(R.drawable.ic_launcher_foreground);
    builder.setContentTitle("Sample Notification");
    builder.setContentText("This is a sample notification");
    builder.setAutoCancel(true);
    builder.setSubText("Tap to view");
    builder.setContentIntent(pendingIntent);

    Notification notification = builder.build();
    NotificationManager mgr = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
    mgr.notify(NOTIFY_ID, notification);
}
```

Intent Resolution Process

- An Intent describing a desired operation
- IntentFilters which describe which operations an Activity can handle
 - Specified either in `AndroidManifest.xml` or programmatically
- Intent Resolution Data
 - Action
 - Data (both URI & TYPE)
 - Category

Specifying Intent-Filter in Manifest

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.intentdemo">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".AppsActivity"></activity>
        <activity android:name=".MediaActivity" />

        <activity android:name=".DestinationActivity">
            <intent-filter>
                <action android:name="android.intent.action.SEND"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <data android:mimeType="*/*"/>
            </intent-filter>
        </activity>

        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

    </application>

</manifest>
```

See: <http://developer.android.com/guide/components/intents-filters.html>

Receiving Implicit Intents

- Note: to receive implicit intents an Activity should specify an Intent-Filter with the category
 - "android.intent.category.DEFAULT"

Android Permission (1)

- Android protects resources and data with permissions
- Used to limit access to:
 - User Information : e.g. Contacts
 - Cost-Sensitive API's : e.g. SMS/MMS
 - System Resources : e.g. Camera, Telephone
- App permissions were declared in `AndrodManifest.xml`

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="codetutor.youtube.edu.contentproviderdemo">

    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_CONTACTS"/>
```

See: <http://developer.android.com/reference/android/Manifest.permission.html>

Android Permissions

Normal

Android grants permissions by default

No need to ask run permissions

INTERENT, NFC, BLUETOOTH,
FINGER_PRINT, ...

Just need to declare in the
AndroidManifest.xml

Dangerous

Android don't grants permissions by default

Must ask run time permissions

PHONT_CALL, CAMERA, LOCATION,
CONTACTS, SMS, MICROPHONE, ...

Besides, Manifest declaration,
additional run time permission
grant is needed.

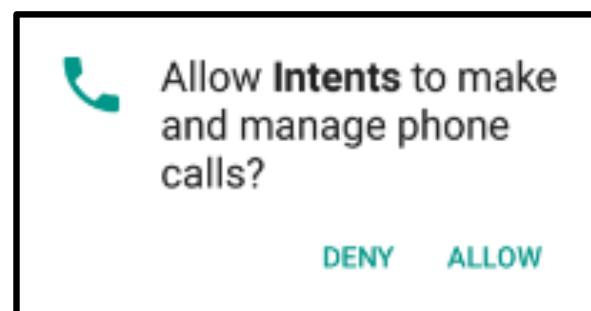
Seek for Run Time Permission

- AndroidManifest.xml

```
<uses-permission android:name="android.permission.CALL_PHONE" />
```

- MainActivity.java's onCreate() method:

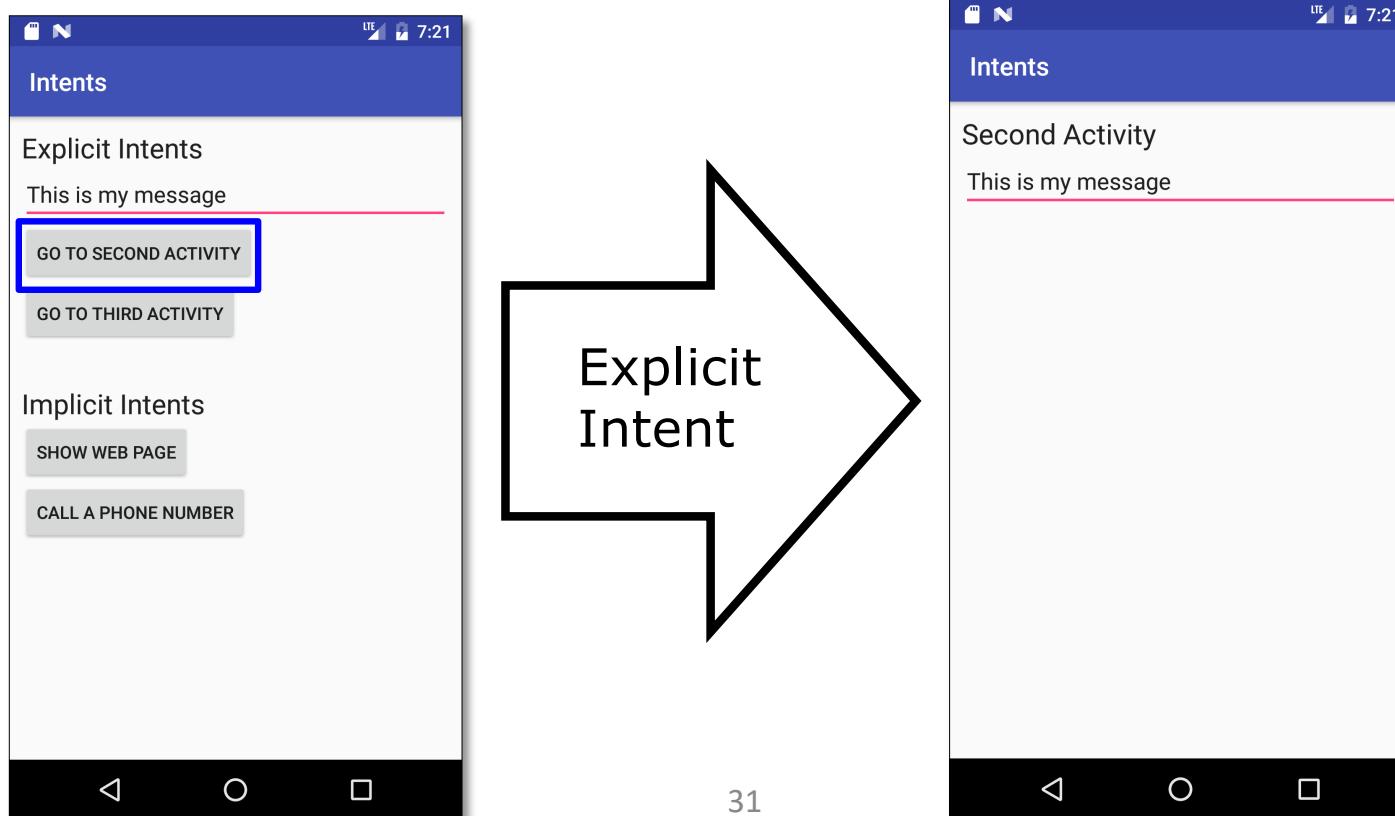
```
// Checking and Seeking for CALL_PHONE permission
if (ContextCompat.checkSelfPermission
    ( context: this, Manifest.permission.CALL_PHONE) != PackageManager.PERMISSION_GRANTED)
{
    ActivityCompat.requestPermissions
        ( activity: this, new String[] {Manifest.permission. CALL_PHONE}, REQUEST_CODE);
}
```



Lab06: Intent Apps

- **Explicit Intent**

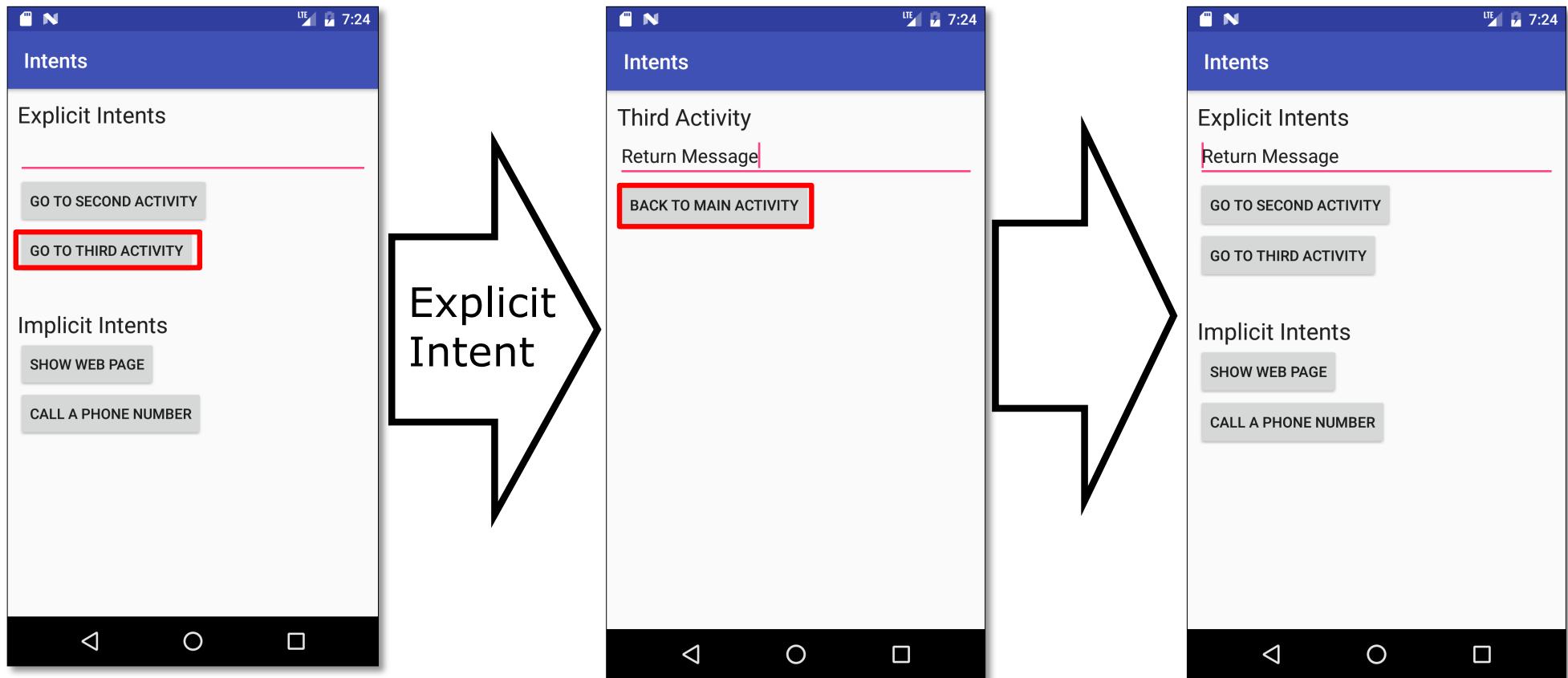
```
Intent intent = new Intent(this, SecondActivity.class);
Bundle bundle = new Bundle();
bundle.putString("Message", myMessage);
intent.putExtras(bundle);
startActivity(intent);
```



Explicit Intent: startActivityForResult()

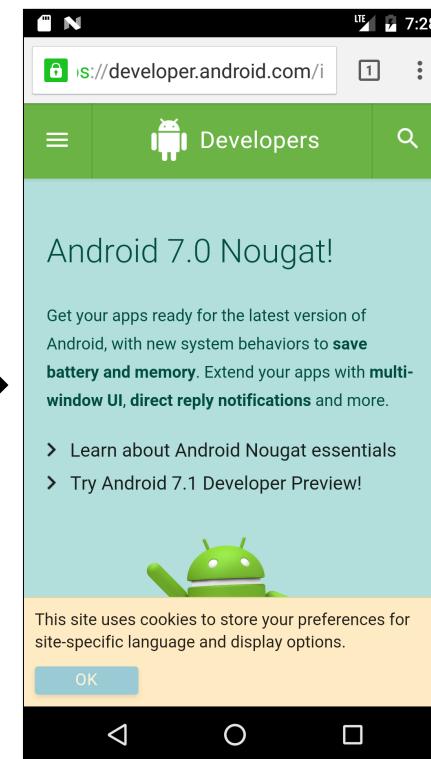
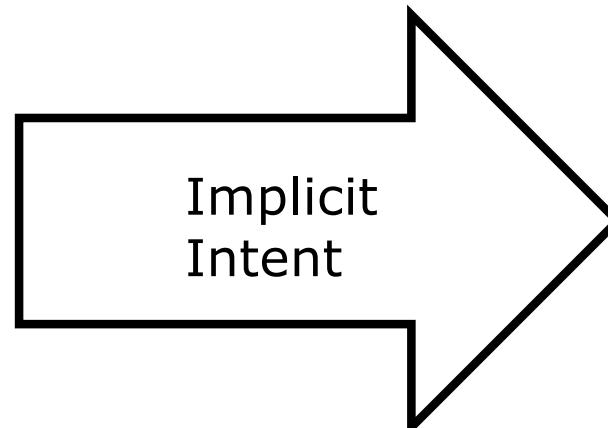
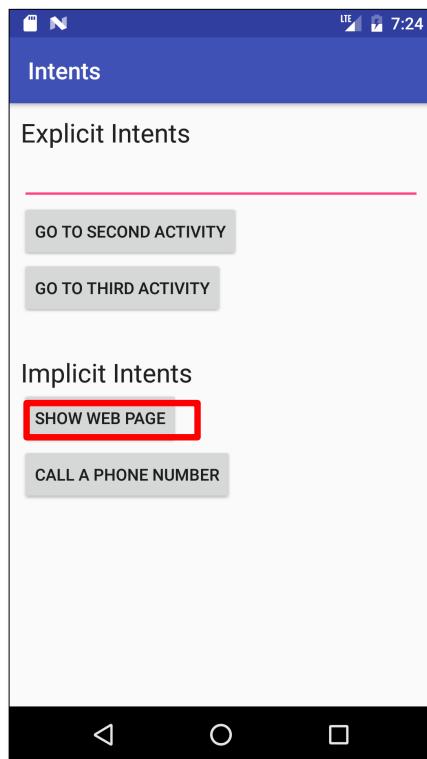
```
public void goToThirdActivity(View view) {  
  
    Intent intent = new Intent(this, ThirdActivity.class);  
    startActivityForResult(intent, requestCode);  
  
}  
  
public void onActivityResult(int requestCode, int resultCode, Intent data) {  
  
    if (requestCode == requestCode) {  
        if (resultCode == RESULT_OK) {  
            et.setText(data.getData().toString());  
        }  
    }  
}
```

Explicit Intent: startActivityForResult()



Implicit Intents (1)

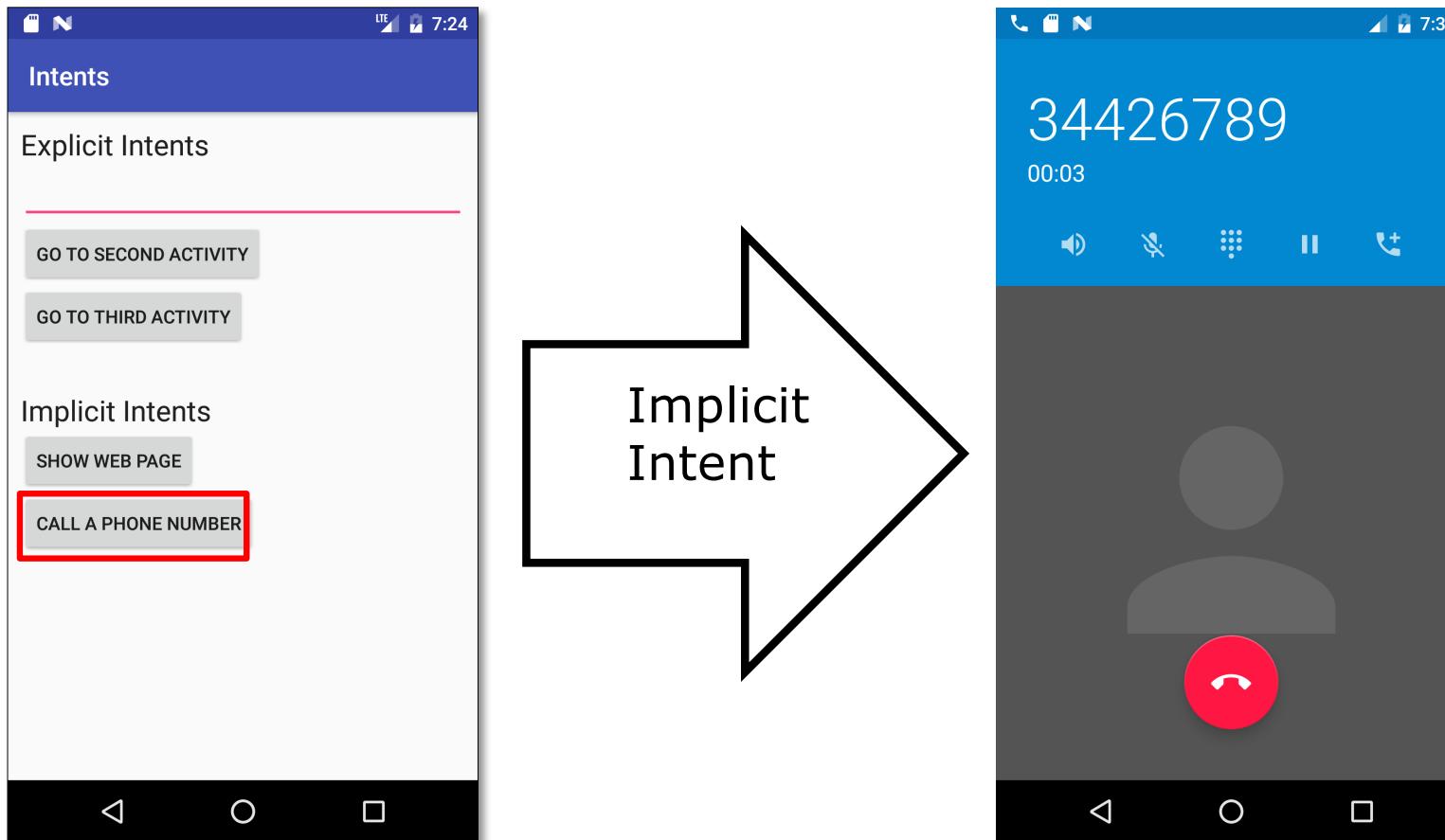
- When the Activity to be activated is not explicitly named, Android tries to find Activities that match the Intent
- This process is called intent resolution



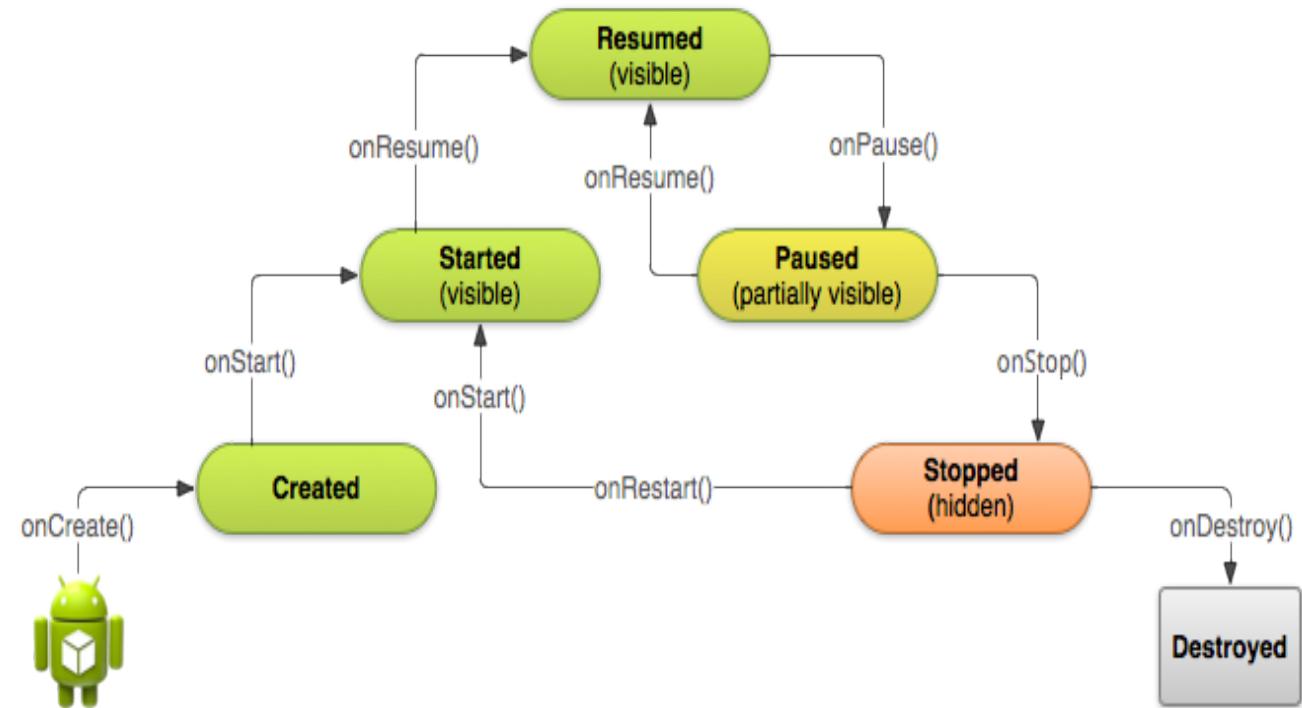
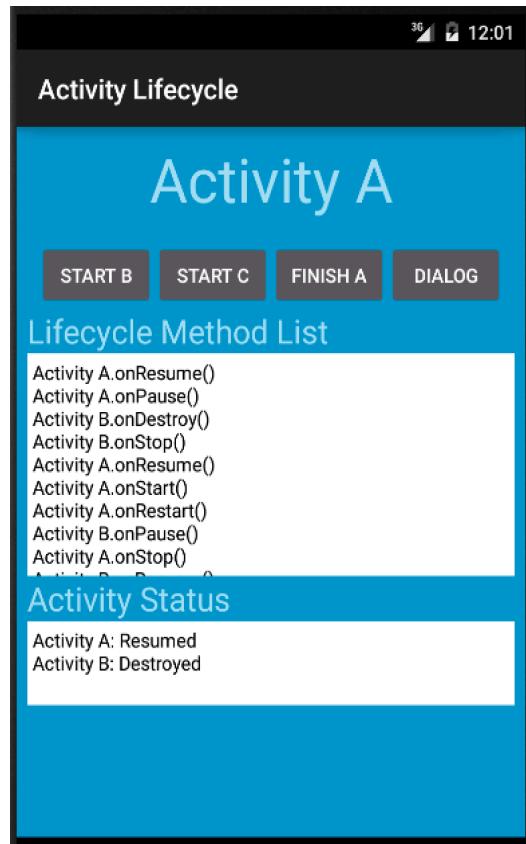
```
Intent intent = new Intent(Intent.ACTION_VIEW,  
                           Uri.parse("http://developer.android.com"));
```

Implicit Intents (2)

```
Intent intent = new Intent(Intent.ACTION_CALL,  
                           Uri.parse("tel:34426789"));
```



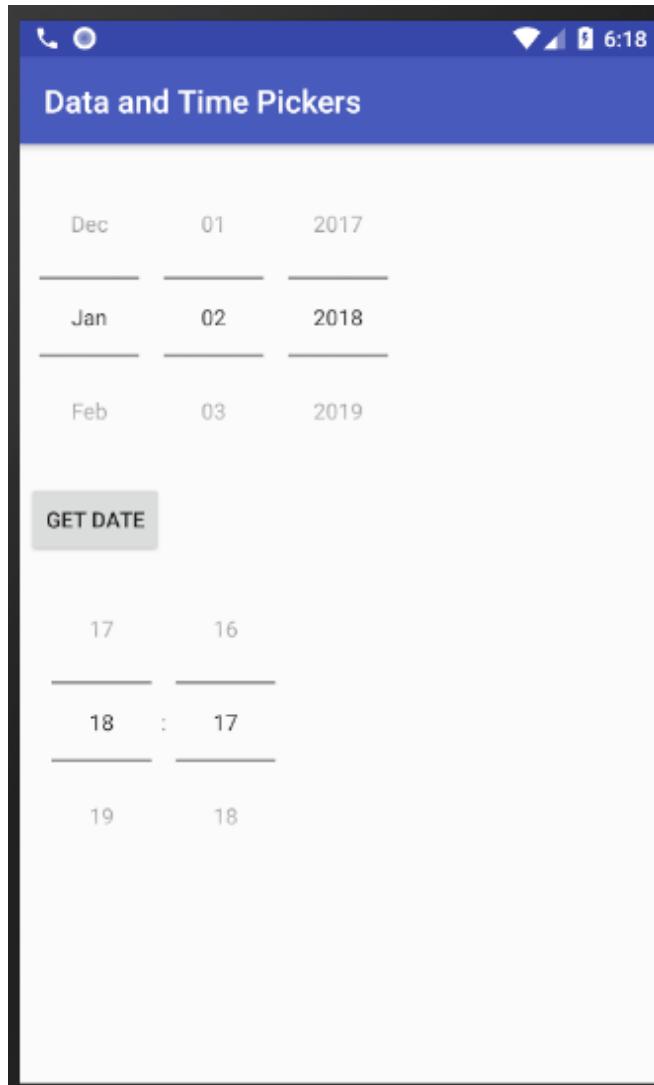
Lab06: Activity Lifecycle App



Training class: <http://developer.android.com/training/basics/activity-lifecycle/index.html>

Demo App Download:
<http://developer.android.com/shareables/training/ActivityLifecycle.zip>

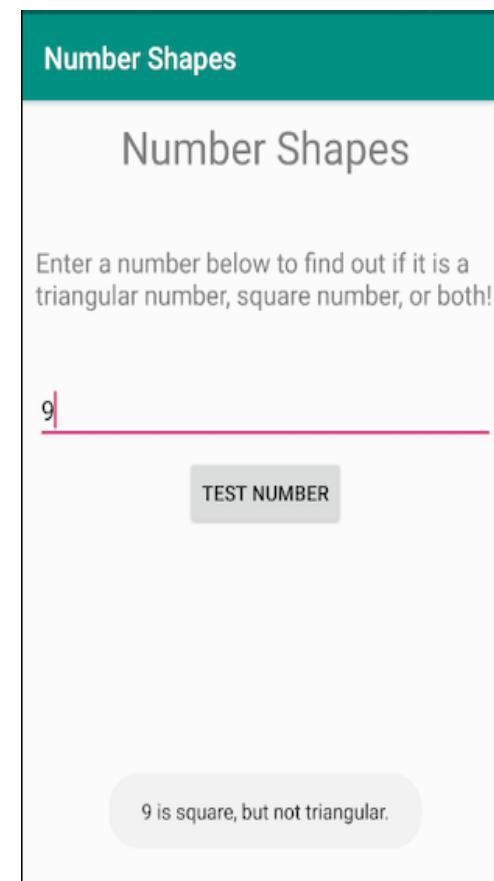
Lab06: Data and Time Pickers App



Lab06: Triangular and Square Testing App

- Mode Class is provided for testing triangular and square numbers:

```
class Number {  
  
    int number;  
    public boolean isSquare() {  
        double squareRoot = Math.sqrt(number);  
        if (squareRoot == Math.floor(squareRoot)) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
  
    public boolean isTriangular() {  
        int x = 1;  
        int triangularNumber = 1;  
  
        while (triangularNumber < number) {  
            x++;  
            triangularNumber = triangularNumber + x;  
        }  
  
        if (triangularNumber == number) {  
            return true;  
        } else {  
            return false;  
        }  
    }  
}
```



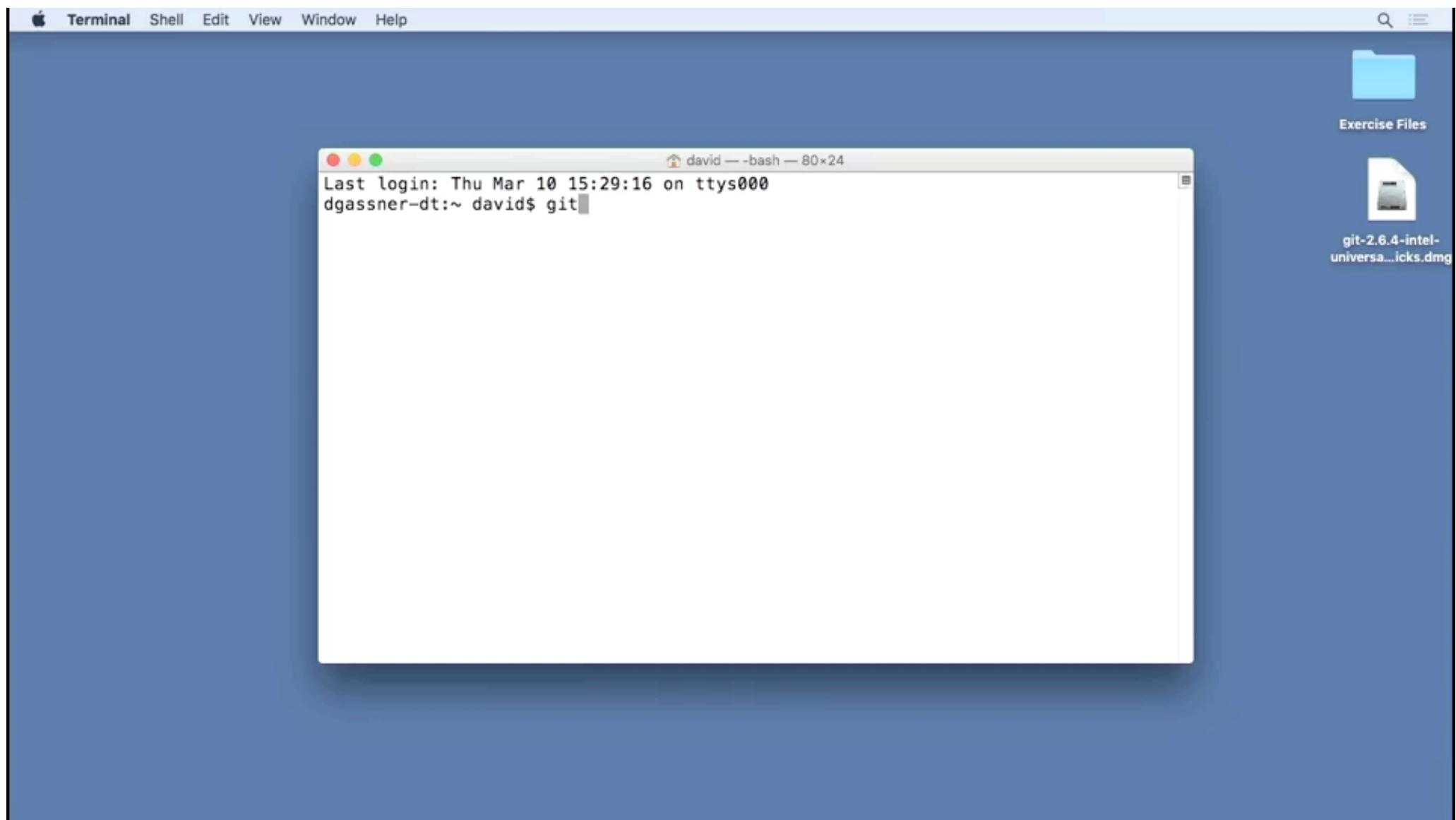
Software Version Control: Git and GitHub

Software Version Control

Git and GitHub

- Install Git on macOS and MS-Windows Platforms
- Configuration Git with your credentials
- Share an Android project on GitHub
- Manage multiple branches with Git

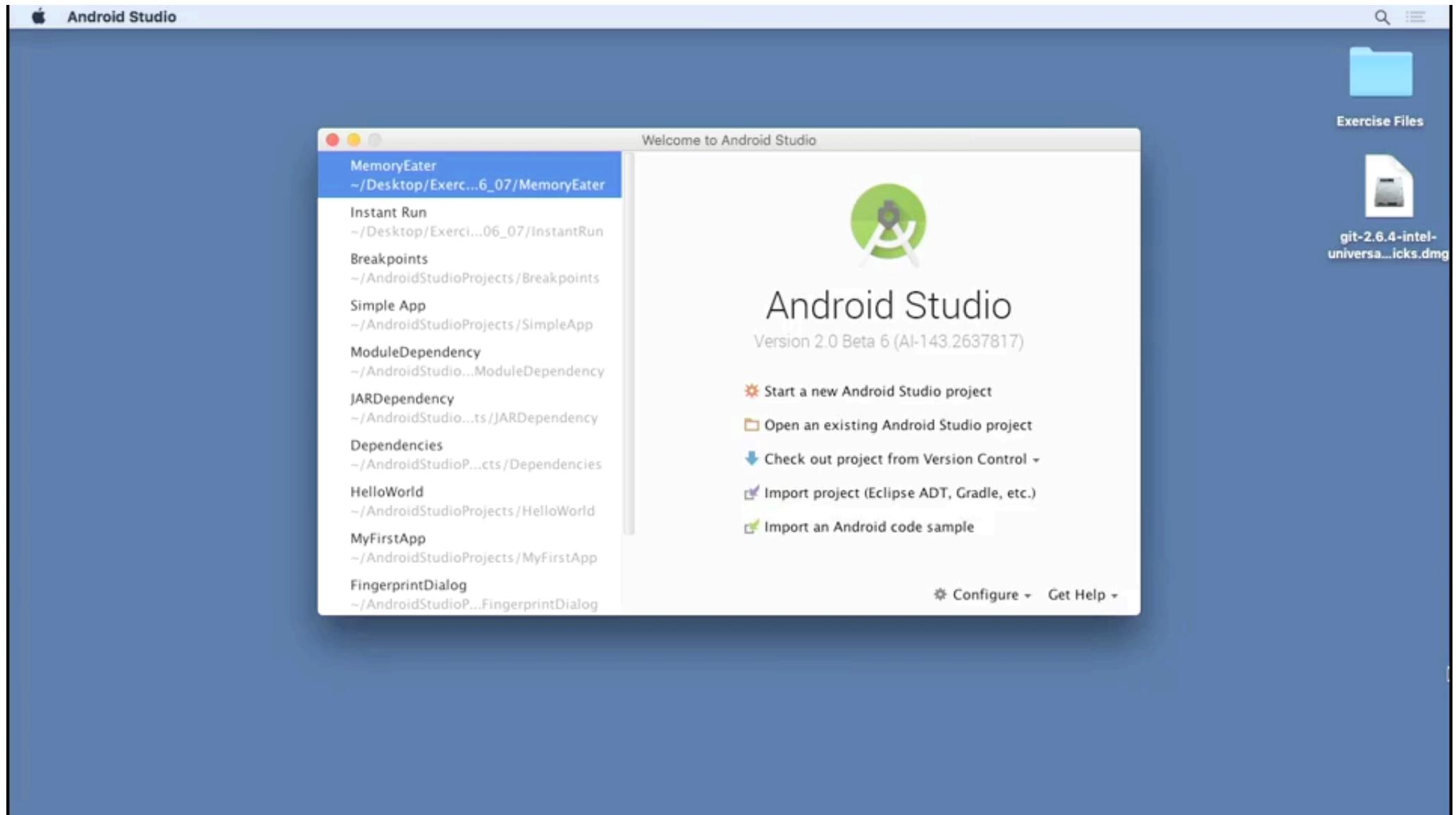
Install Git on macOS



Install Git on MS-Windows



Configure Git with your credentials



Share an Android Project on GitHub

A screenshot of a web browser window showing the GitHub homepage. The browser's top bar includes the title 'GitHub · Where software is built', the URL 'GitHub, Inc. [US] https://github.com', and a user profile for 'David'. The GitHub logo is at the top left, followed by a search bar and navigation links for 'Explore', 'Features', 'Enterprise', and 'Pricing'. A green 'Sign up' button is on the right, and a 'Sign in' link is just below it. The main content area features the tagline 'Where software is built' in large white text, followed by a description of GitHub's collaboration and code management features. It also mentions that public projects are free and private plans start at \$7/mo. On the right side, there is a sign-up form with fields for 'Pick a username', 'Your email', and 'Create a password', along with a note about password requirements and a large green 'Sign up for GitHub' button. At the bottom, there is a blue banner with the text 'Want to use GitHub on your servers?'

Want to use GitHub on your servers?

Mange Multiple Branches with Git

