

CSC 4360/ CSC 6360

Mobile App Dev
Welcome!

Dr. Erin-Elizabeth A. Durham



- PhD in Computer Science from Georgia State University (2014)
- > 15 years in private industry, working directly in Computer Science



- Happily married, mother of 4 great kids
- Regular employment: General Electric, Power Technology
- Evenings: Teaching at Georgia State University, Computer Science Department (*Mobile App Dev, Big Data Programming, Databases, Software Development, etc*)



- HackGSU faculty/staff sponsor

CSC 4360/ CSC 6360

Mobile App Dev
LAB 1

Items of Note

- **Required textbook:**

Mobile App Development for iOS and Android, Edition 2.0
 Jakob Iversen & Michael Eierman
 Published by Prospect Press, 2017
 ISBN-10: 1943153280
 ISBN-13: 978-1943153282
 eTextbook option
 ISBN-13: 978-1943153275
 ISBN-10: 1943153272

- Online version:
<https://www.vitalsource.com/referral?term=9781943153275>
- Github source:
<http://github.com/LearningMobile/BookApps2.0>

Items of Note

- Syllabus is online (D2L / iCollege)
- We use D2L/iCollege a LOT in this class. Check your calendar!
- Slides are posted AFTER class to D2L/iCollege.
- Attendance roster signed at every class session. It counts for a **grade**, therefore do NOT sign for other students. Ever.
- Yes, there will be a group project. No you cannot do the team project by yourself. Your number is your TEAM. Teams are assigned already – you will be able to see your team in D2L/iCollege.

Class D2L/iCollege Homepage

- Assignments
 - Description
 - Due Dates
 - Upload folder(s)
- Calendar of events
 - Class Activities
 - Assignments
 - Quizzes/Test Dates
- Grade Book
 - TA uploads your grades directly to D2L

If you cannot get onto the class D2L/iCollege site, please let me know immediately in class today

Grading Rubric

- Does the program run? (apk, etc)
- Does the program give the correct/expected output?
- Does the program meet the listed requirements of the assigned problem?
- Code quality:
 - Named per directions in Homework or Lab assignment
 - Clean, clear structure, good quality
 - Appropriately commented
 - Documented (readme.txt, run instructions, etc)

Grading Policy

- You will have until midnight on the submission date to turn in your assignment.
- **You can submit your assignment(s) early.**
- Late assignments can only be submitted until the Late deadline (usually before the next class period starts – **ALWAYS** verify on D2L/iCollege).
 - Late assignments will receive an 10% automatic penalty.
- Any updates to the assignment/due date will be published on the class D2L site **ONLY**

What's first in our class

- Android Development
<https://developer.android.com/studio/index.html>
 - Open-source O/S written in Java and *executed* with interpreters, similar to but not the same as the Java Virtual Machine (JVM)
- Android is relatively easy to port to any hardware platform.
 - Android was designed for mobile devices, so it is nearly always run on them, rather than on stationary computers, which have limited mobility.

Software in this class

- First homework will use android studio
- **Everyone** will install android studio
- Later in the semester, we will work on iOS, Windows, and other platforms. We will use VMs for students who do not have an Apple device for the iPhone development

What Next?

- D2L skills survey results, Homework #1
- Purchase your textbook. If you cannot get it at the bookstore, it is available on Amazon
Mobile App Development for iOS and Android, **Edition 2.0**
Jakob Iversen & Michael Eierman
Published by Prospect Press, **2017**
ISBN-10: 1943153280
ISBN-13: 978-1943153282
eTextbook option
ISBN-13: 978-1943153275
ISBN-10: 1943153272
Online version:
<https://www.vitalsource.com/referral?term=9781943153275>

Textbook source code

Source code for your textbook "Mobile App Development for iOS and Android" by Michael Eierman and Jakob Iversen

<https://github.com/LearningMobile/BookApps2.0>

You cannot use the textbook source code directly in a homework / lab problem UNLESS specified in the homework / lab directions. You are WELCOME to view how the authors solved the problem, and do something similar :)

We will occasionally reference this code directly!

Install Android Studio

<https://developer.android.com/studio/index.html>

Already installed/done? You can either:

- 1) Work on Homework #1
- 2) Take the free Udacity github course:
<https://www.udacity.com/course/how-to-use-git-and-github-ud775>

(Please wear headphones)

Getting Started

Architecture of Android Applications

- All Android applications use the MVC architecture
- **Activities**
 - Related to *servlets*
 - An activity is a Java class instance with an associated view markup file
 - Each activity manages one page of display

Architecture of Android Applications

View Files

- XML documents that display forms

Intents

- A means of communication within and among applications
- An intent is a Java class instance, often used to start and stop activities

Architecture of Android Applications

Implementation

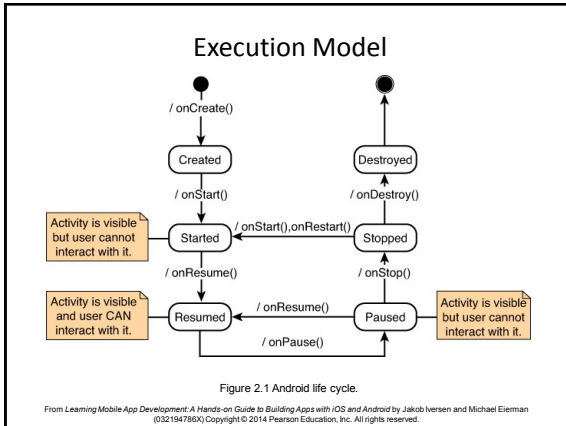
- An Android application is built as a project, with one or more activities

Dalvik VM

- Applications are compiled to an intermediate form similar to byte code, interpreted by the Dalvik VM
- Each application runs in its own process on its own copy of the Dalvik VM

Execution Model

- An Android application starts execution when the user starts it
- It ends when the user starts another application or the operating system stops it to capture its resources
- Execution is controlled by callback methods that are implicitly called when system or user-raised events occur
- Activities are placed on a stack, with the top one currently running with its view displayed



Execution Model

- The `onCreate` method sets the activity's view and often initializes some class-scope variables
- The `onStart`, which is implicitly called when `onCreate` is finished, displays the view and calls `onResume`
- The `onResume` starts the execution of the activity's code

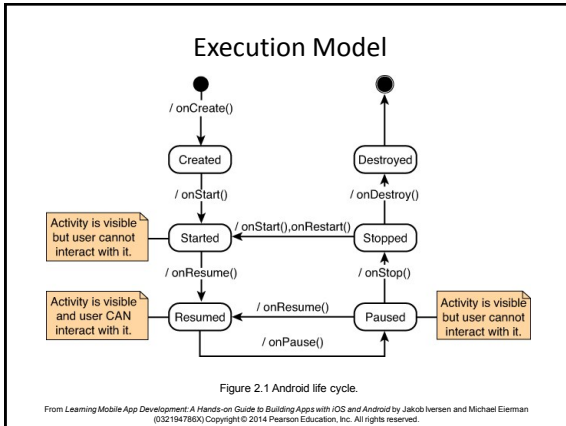
Execution Model

The `onPause` method takes an activity from the *resumed* state to the *paused* state

The `onStop` method takes an activity from the *paused* state to the *stopped* state

The `onDestroy` method takes an activity to the *destroyed* state


All of these are called the *life-cycle methods*



An aside...

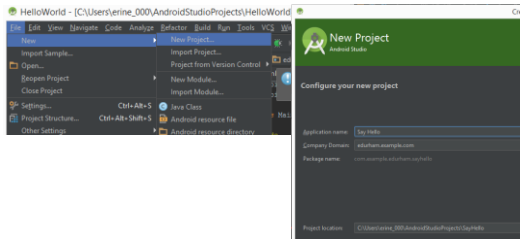
- The Execution model for Android is VERY different from the iOS model (covered later in the semester)
- Yes, being able to IDENTIFY the differences between the Android and the iOS execution model(s) will be a TEST question on the Midterm.

Your first application

- Open Android Studio  **Android Studio**

New Project, “Say Hello”

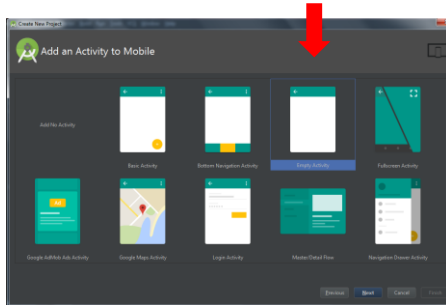
- Start a new project, and name it “Say Hello”



Choose your Target Devices

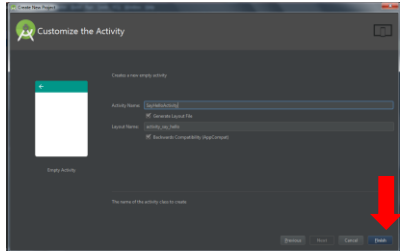
- Asides:
 - Lower SDKs have a larger set of devices that can use them
 - Higher SDKs have more features
 - You will have to download the SDK you choose, if it is not already on your computer
- Demo

Add an Empty Activity

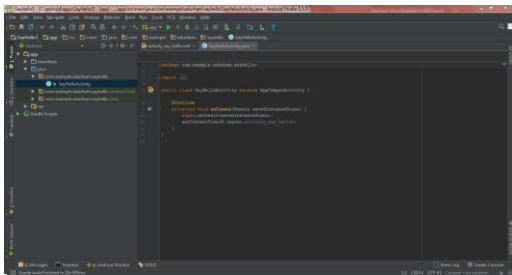


Create a new blank activity

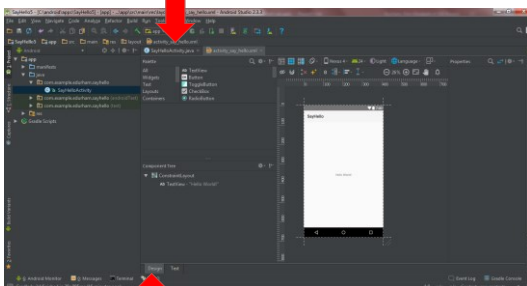
- Create a new blank activity and call it SayHelloActivity (or similar)



Initial

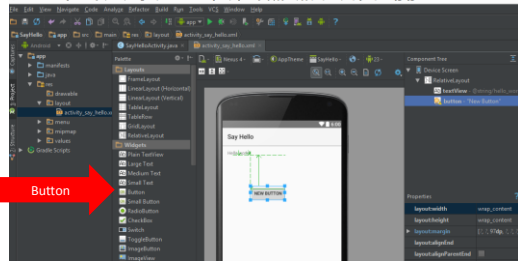


Click xml file (tab) and "Design"



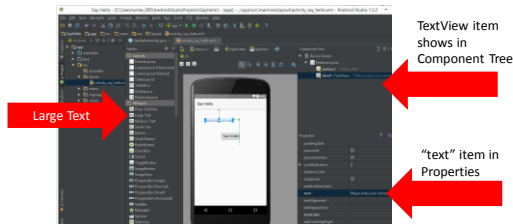
Drag a button onto the screen

- Change the text to “Talk to me”



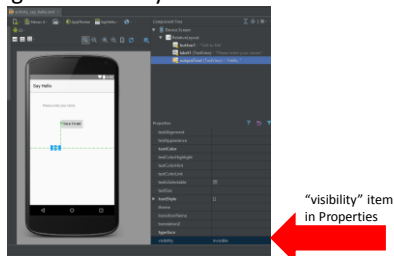
Add a label (TextView item)

- Change the text to “Please enter your name” and the id to “label1”



Add another textview for the output

- Name the element “outputText”
- Change the visibility to “invisible”



Add an “EditText” to accept input

- Name your element “inputText”



Edit the java code

- Go to your SayHelloActivity.java file and add elements to work with the text input and output

```
public class SayHelloActivity extends Activity {
    private EditText inputText;
    private TextView outputText;
}
```

- Find the elements and assign values, i.e.
`yourName = (EditText) findViewById(R.id.inputText);`
`outputName = (TextView) findViewById(R.id.outputText);`

SayHelloActivity.java

```
package com.example.android.hello;

import android.support.design.widget.Snackbar;
import android.support.design.widget.TextInputLayout;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.design.widget.TextInputEditText;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class SayHelloActivity extends AppCompatActivity {

    private EditText inputText;
    private TextView outputText;

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

        inputText = (EditText) findViewById(R.id.inputText);
        outputText = (TextView) findViewById(R.id.outputText);

        Button btnHello = (Button) findViewById(R.id.button_hello);
        btnHello.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String yourName = inputText.getText().toString();
                String outputName = outputText.getText().toString();
                outputText.setText(yourName + " Hello " + outputName);
            }
        });
    }
}
```

Final code should look like:

```
public class SayHelloActivity extends Activity {
```

```
    private EditText yourName;  
    private TextView outputName;
```

```
    public void printHello(View v) {
```

```
        Button helloButton = (Button) v;  
        ((Button) v).setText("You clicked me!");
```

```
        yourName = (EditText) findViewById(R.id.inputText);
```

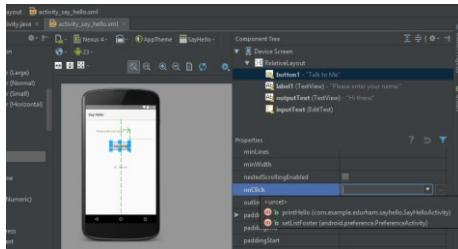
```
        outputName = (TextView) findViewById(R.id.outputText);  
        outputName.setText("Hello, " + yourName.getText());  
        outputName.setVisibility(View.VISIBLE);
```

```
    }
```

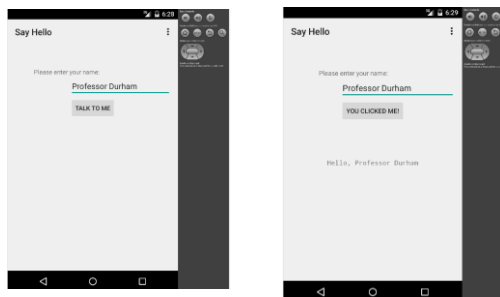
```
...
```

Go back to the Design Screen

- Add printHello to the button's onClick



Run It in the Emulator and TEST!



Want to see it again, in detail?

- Good resource:

<https://www.youtube.com/watch?v=OWL0MCvtSC8>
