EECS1022
MOBILE COMPUTING

COURSE OVERVIEW

(SLIDES ADAPTED FROM PROF.H. ROUMANI)

PROF. Y. LESPÉRANCE
Dept. of Electrical Engineering & Computer Science

TOPICS & TECHNOLOGIES

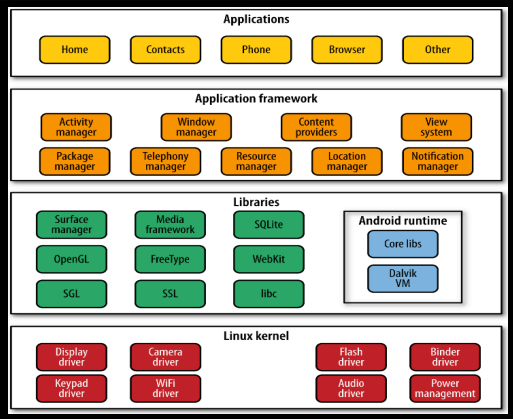
- Abstraction & Separation of Concerns
- The Software Development Cycle
- Object Oriented Programming [OOP]

- Android App Development
- User Interface [UI] Design
- The Java Programming Language

APPROACH

- Builds on EECS1012
  Separation of Concerns, Computational thinking.
- Industrial-Strength Tools
  UI via XML (*not* HTML), Behavior via Java (*not* JS).
- Solid Platform
  O/S is Android, IDE is Studio.
- Experiential Pedagogy
  Foundational concepts in class + real-life projects in lab

ANDROID: THE STACK



Android Versions [A/B in 2008]

XML

- Like a Generalized HTML
  But 100% strict.
- Well-Formed XML
  Obeys the syntax rules:
  See: http://www.w3schools.com/xml/xml_syntax.asp
- Valid XML
  Must be well-formed and obeys a schema that dictates the names of tags and attributes (namespace) and sets the types of their values.

XML Example –An Android Layout

```
<LinearLayout layout_width="match_parent" orientation="vertical">

  <EditText layout_width="match_parent" id="width"/>
  <EditText layout_width="match_parent" id="height"/>

  <Button layout_width="match_parent" text="Compute"  id="button"/>

</LinearLayout>
```

Locate:
Document root, tag, closing tag, attribute, attribute value

Note the naming style for multi-word identifiers:
Pascal, Camel, or underscore.

Java

- Adopts the C Syntax
  Same as JavaScript
- Strongly-Typed
  Syntax errors exposed as you type. Static checking of potential runtime and logic errors.
- OOP
  Programming by Delegation.
- Platform-Independent
  Write once, run anywhere.

Java Example –A Class

```
public class Rectangle
{
   private int width;
   private int height;

   public Rectangle(int w, int h)
   {
      this.width = w;
      this.height = h;
   }

   public int getArea()
   {
      int result = this.width * this.height;
      return result;
   }
}
```

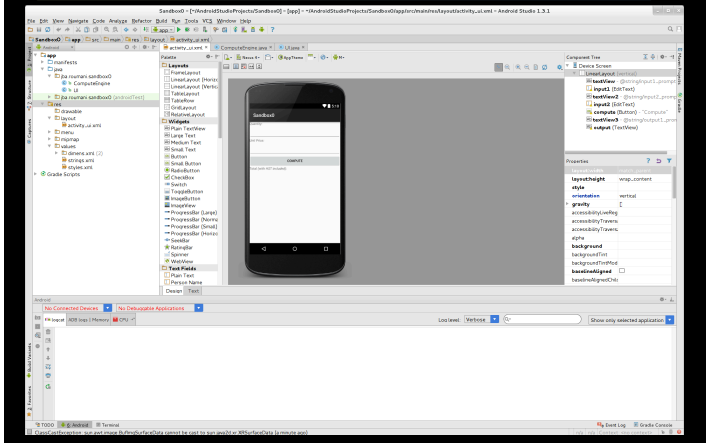
- *Class*
- *Block*
- *Attributes*
- *Constructor*
- *Method*
- *Method Return*
- *Parameter*
- *Variable*
- *Declaration*
- *Assignment*
- *Identifier*
- *Keyword*
- *Operator*
- *Separator*
- *Literal*

Android Studio

Makes writing code easier (compile-as-you-type); designing UI easier (drag widgets, set properties); running and debugging.

- Start your Vbox
  Open a terminal a note the AndroidStudioProjects folder
- Launch Studio
  From the *Application*, *Programming* menu.
- Start a New Project
  Programming by Delegation.
- Project Location
  /home/user/AndroidStudioProjects/

The Studio Ecosystem



The Studio Ecosystem