Topics in Computer Science: Mobile Application Development

Lecture 1

Prof. El-Sheikh

CS-591, MOBILE APP DEV.

Topics

- Android Market Penetration.
- Why we are targeting Android 5.0 and not Android 6, 7 or 8 (Android Fragmentation)
- Review of Syllabus
- Ice Breakers
- Design Task - Worksheet 1
- Discussion of third party collaborations.
- Share follow up info from students regarding employment.
- How the class is different, very hands on, grounded in industry best practices.
- Drop, if you don't have time.
- Attendance and Participation Counts.
- Pace of class is swift.
- It's a seminar and you will be expected to pursue quite a bit of research on your own, but I will ground the research for you.
- You will also be expected to work well with your peers. Be nice!

Quick Overview of Today

Overview Lecture 1

- Adminstrivia
- Course Overview
- Introductions/Ice Breakers
- Statistics
  - Market Penetration, Android vs. iOS vs. Win vs. Bb…
  - Market Fragmentation
- Syllabus
- Course expectations
- Course Layout
  - 2 halves
- Story Boarding
- Brainstorming Apps (Worksheet)
- How does Android Studio's IDE Stackup?
  - *Not so great, but it's getting better.*
  - *Comparisons with other IDE's, eg, Visual Studio*

Overview Lecture 1 *(cont.)*

- Virtual Machines
  - The Dalvik VM (DVM)
  - v. the JVM v. NET
- Why are some Android apps twice baked?
- Are modern Android apps also twice baked?
- Android Market Fragmentation
- Historical perspective of Android, tying into the “Java Craze” of the 90’s.
- Worksheet
- Will Create a Test App and Compare/Contrast with other IDE’s.

ADMINISTRIVA

Administriva & Reminders

- Bring your Laptops,
  - Strongly recommend getting a physical device, they are cheap now.
  - Run the emulator (if you need it) as soon as you arrive. ☺
  - **Please target Android Lollipop, 5.x**
- Reminder: Attendance, Collaboration and Participation are mandatory.
  - Use Piazza to help each other.
    - **Small amounts of extra credit for those who post often, esp. those who provide "good" answers.**
  - Will drop lowest assignment.
  - Don't miss Midterm or other important classes.
  - Course is highly collaborative. **No Makeups.**
  - **If you are overloaded or have limited time please drop and retake when you have more availability.**
- Feel free to work in groups on all assignments
  - You must specify clearly who you worked with.

Textbook

- Bookstore should have it.
- Jones & Bartlett Website, to purchase directly.
  - http://www.jblearning.com/catalog/9781284092127/
- Use code BUANDROID, 30% off.
  - Anyone tried, does it work?
- Android App Development Herv Franceschi
- ISBN-13: 9781284092127
- Jones & Bartlett Learning;

Other Expectations

- Be ready to commit significant hours to this course.
  - Lots of new material + independent research.
  - It's a seminar course in the truest sense.
  - Android Studio is an immature and "clunky" be patient with this IDE.
- Be considerate.
  - Do not use electronics while I am lecturing. Or while your peers are presenting.
    - Doing so will effect your participation grade.
- Be accommodating.
  - You will be working very closely with your team.
  - Be flexible, polite, and helpful.
- Have High Expectations – Of your team.
  - No one should be coasting.
  - If members of your team are not contributing adequately, press them (politely).
  - Then come see me to discuss.
  - I will be checking in frequently.

Introductions & Ice Breakers

About Me

- Dual Background in Electrical and Computer Engineering & Education
- For the past 20 years I have been trying to square these two disparate fields.

- Try whenever possible to bring real world experience into the classroom.
  - I do come from Corporate America
- Educational Philosophy:
  - Strong Belief in Collaboration and ongoing assessments – not a big believer in high stakes testing, even at this level.
  - Homework philosophy
    - It's not cheating if you call it collaboration
    - Balancing homework with ongoing assessments ensures students are on task.
- Love what I am currently doing,
  - Splitting time between BU and Consulting.
    - Consulting for the State of Maine, CDC
    - Also working with a local NonProfit

About Me

- Corporate Experiences (primarily as software consultant)

Organization	Sector
MITRE/US Army	FFRDC, DOD
Harvard School of Public Health	Academia
Computer Sciences Corporation/NHTSA	DOT – NHTSA
Aramark	Corporate/POS
Martin Dawes Analytics	Corporate/BI
Linedata Services	Financial
Sovos Compliance	Financial
Philips Healthcare	Medical *(briefly)*
Government of the State of Maine – CDC	DHHS *(Current)*
Boston Building Resources	Non Profit *(Current)*
SAIC/Federal Govt.	DOT – FMCSA *(Current)*

Ice Breaker

- Introductions/Ice Breakers
  - One thing about your home town
  - What you hope to gain from the course
  - Do you have any ideas for a project already?

Statistics

- Statistics
  - Market Penetration, Android vs. iOS vs. Win vs. Bb…
    - http://www.kantarworldpanel.com/global/smartphone-os-market-share/
    - https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1&qpsp=2011&qpnp=1&qptimeframe=Y
- Market Fragmentation
  - Easiest way to see this is by creating a new project in the IDE,

Let's try it.

Course Overview

Syllabus

- Course Objectives
- Project Based, Attendance and Participation is Mandatory.
- Course Grading Criteria
- Course Approach
  - Didactic at first
  - Research based later
    - Depends on the technologies you want to use in your final project.
- Final Project
  - Incorporate the things you have learned from the first half of course, including:
    - consumption of hardware sensor data, eg, gyroscope, compass, accelerometer
    - then choose from a “Buffet” of Technologies, with at least two advanced topics that you will research with your team. (Eg, Facebook/Twitter/Google Maps/Instagram/VLC/etc.)

What you will learn in this course

- How to develop Android Applications using Android Development Studio and Java.
- Event Driven UI Programming.
- The parts of an Android Application.
- Life Cycle of an Android Application
- Activities, Fragments, Intents, Database Development, File I/O, GPS/Google Maps, Consumption of Sensor Data.
- We will occasionally demonstrate concepts in other development environments, to help you with future pivots. *Including Today.* ☺

Course Layout

- Course has two overlapping halves.
  - First half didactic – students learn about developing mobile applications.
    - App Lifecycle
    - Component Dev
    - Event Handlers
    - Interacting with Device HW
    - Interacting with Google Maps
  - Second half project based – students research and implement an App.
    - Will leverage what you learned during first half.
    - Work in teams of ~4.
    - Agile teams.
    - Final Project with Presentations.
    - Mini Research Tasks to help you implement your Final Project.

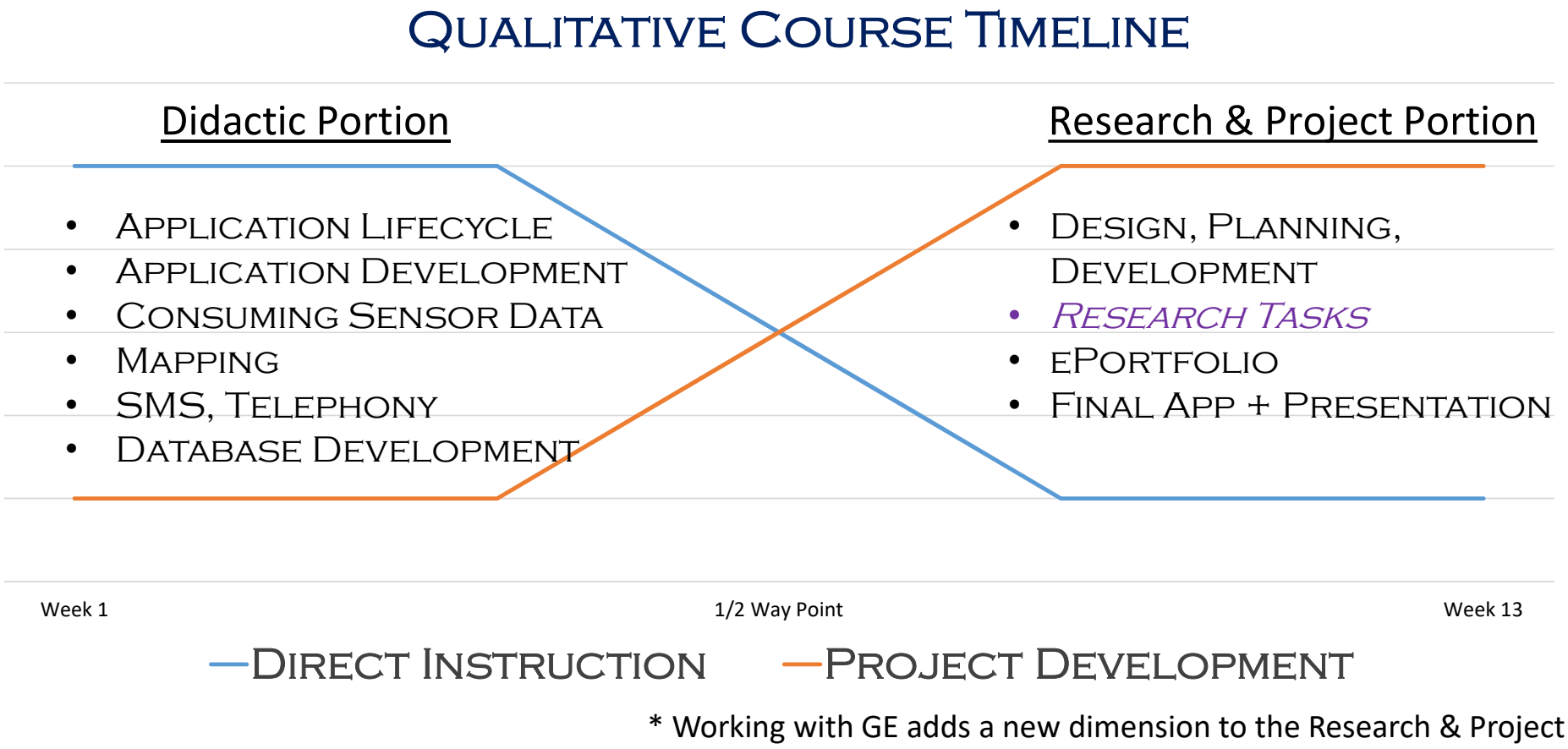
The two parts of the course…

- The first half of course,
  - I will lay the ground work.
  - Each week I will teach you core concepts of Android App. Development.
  - This will include topics on UI development,
  - Event Handling, Gestures
  - Component/Fragment development and reuse
  - Consumption of sensor data, including GPS,
  - Database Development

Course has two halves *(cont.)*

- The second half of the course will be geared towards developing your final project.
  - You will work in agile teams.
  - Use the information I provided you in the first half of course as scaffolding
  - Develop a useful application.
  - Present your work (ongoing presentations ~every 2 weeks)
  - Final Presentation
  - Deploy your application to the Google Play Store ($25)
  - Second half will overlap with first half, giving you a chance to meet your team and develop project ideas.

Managing Time Constraints: Overlapping Halves



* Working with GE adds a new dimension to the Research & Project Portion

Approach

- Android development is a large area.
  - There are two parts to the course.
- The first half of the course will be didactic.
  - I will guide you and teach you specific android concepts
- The second half of the course will be Project Oriented
  - I will continue to guide you, but some of the information needed to complete your project will require you to do some research.
- Attendance, Collaboration and Participation are mandatory.

Approach *(cont.)*

- Learn by example
- Whenever feasible (and time allows) I will demonstrate concepts in other IDE's
  - Eg, an Activity is sometimes called a Form in other IDE's
  - Eg, a Fragment could be considered a reusable Component or Frame in other IDE's
  - Eg, Writing Event Handlers is common across all GUI IDE's, but implementation is different in Java and Android Studio vs. C++, C#, VB, Delphi, etc.
  - Goal is to provide multiple entry points into material.
  - This approach will also help you pivot down the road.

Course Expectations

- You will need to come prepared to each lecture.
  - Partially Flipped Classroom
  - Pre Readings
  - Meetings with your development team
- You will work with in ***agile*** teams of ~5 students (more on this later).
- Everyone must participate and produce equally.
  - 1 week sprints
- You will each take turns as ***Scrum Master***
- I will serve as product owner,
  - You will present your accomplishments after each sprint, and your goals for subsequent sprints.

Pivot

- This course will require you to pivot and utilize your understanding of Java to develop rich Mobile Applications for Android.
- This is an advanced level course.
- You will be required to do your own research during the second half of the course in support of your Final Project.
  - Eg, how to interact with eBay, AWS, Twitter, Facebook.
  - Most of these API's are well documented.

What is *our* Agile Approach to Development?

- Agile approach to development?
  - Work in small teams of 5
  - Every week you will appoint a new Scrum Master
    - Responsible for tracking project progress, initiating and scheduling tasks.
    - Reporting Progress and Planning Subsequent Tasks.
    - This will be a rotating position amongst your team.
- Scrum Master will provide weekly “standups” in lecture, detailing what your objectives were, how you achieved those objectives, any problems or blocking issues, and what the plan is moving forward.

Potential for Collaboration with outside companies and organizations

- Exciting time to take this course!
- Opportunities to collaborate with outside organizations on projects through BU's Incubator Program in the Hariri Center, BU SPARK.
- More on this later.

Homework Philosophy

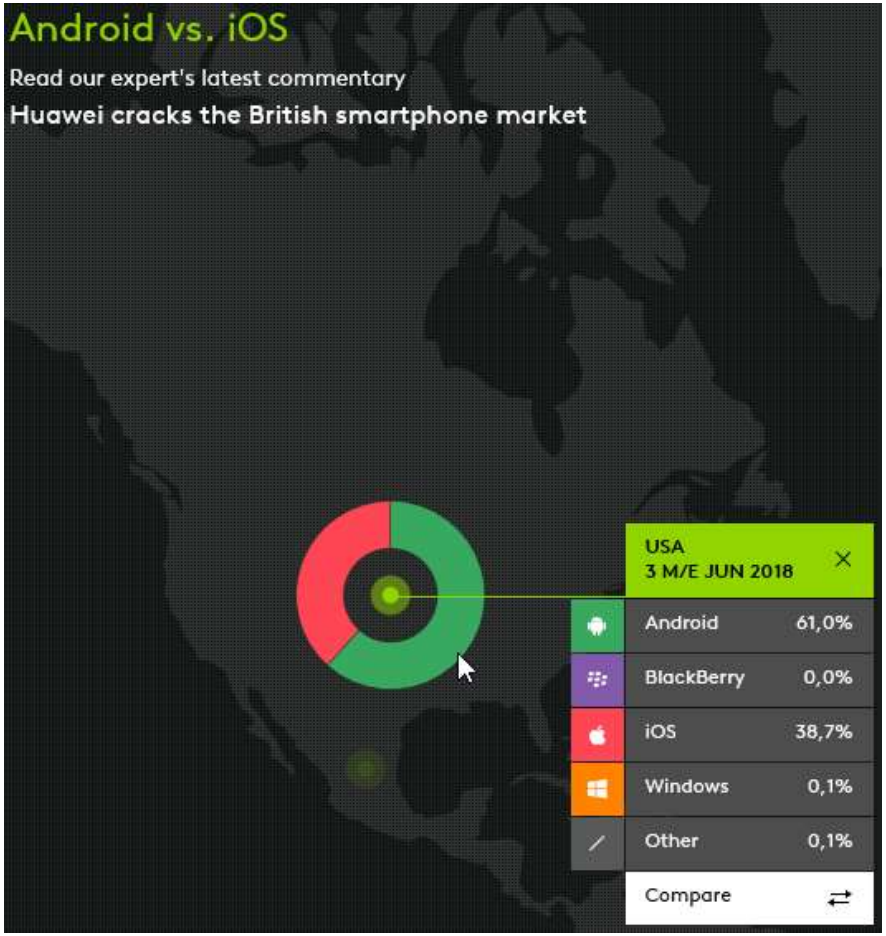
Homework Philosophy

- Big Believer in Collaboration
  - Homework is just another way to collaborate.
- So, why limit students?
- My approach is to allow students great latitude when collaborating, going as far as to allow them to submit jointly, provided they document who they work with.
  - Won't students "cheat", it's not cheating if you call it collaborating.
- Won't some students just "skate"?
  - Nope, not if you closely tie homework to classwork/projects/and other assessments. Nope, not if you provide ongoing assessments and checkin frequently

~~• Javert!~~

Android Market Share

Why Android? By Region 2012-Present



Ref: http://www.kantarworldpanel.com/global/smartphone-os-market-share/

Internet Explorer seems to work best when opening.

Why are we targeting an older version of Android, Lollipop (5.x)?

- With Android there is an issue of market fragmentation.
- As new versions of Android hit the market, users are encouraged to upgrade their hardware, if they can.
- Some hardware just can't be updated, and some users just don't bother.
- **Q: Oreo is the latest, so, why don't more developers target this version.**
- https://developer.android.com/about/dashboards/index.html

Please Target Android Lollipop, 5.x

- For all of Your Assignments, Please Target Android Lollipop, 5.x

- Note: We will discuss specific features of the newer versions of Android.

Brain Storming Apps & Story Boarding

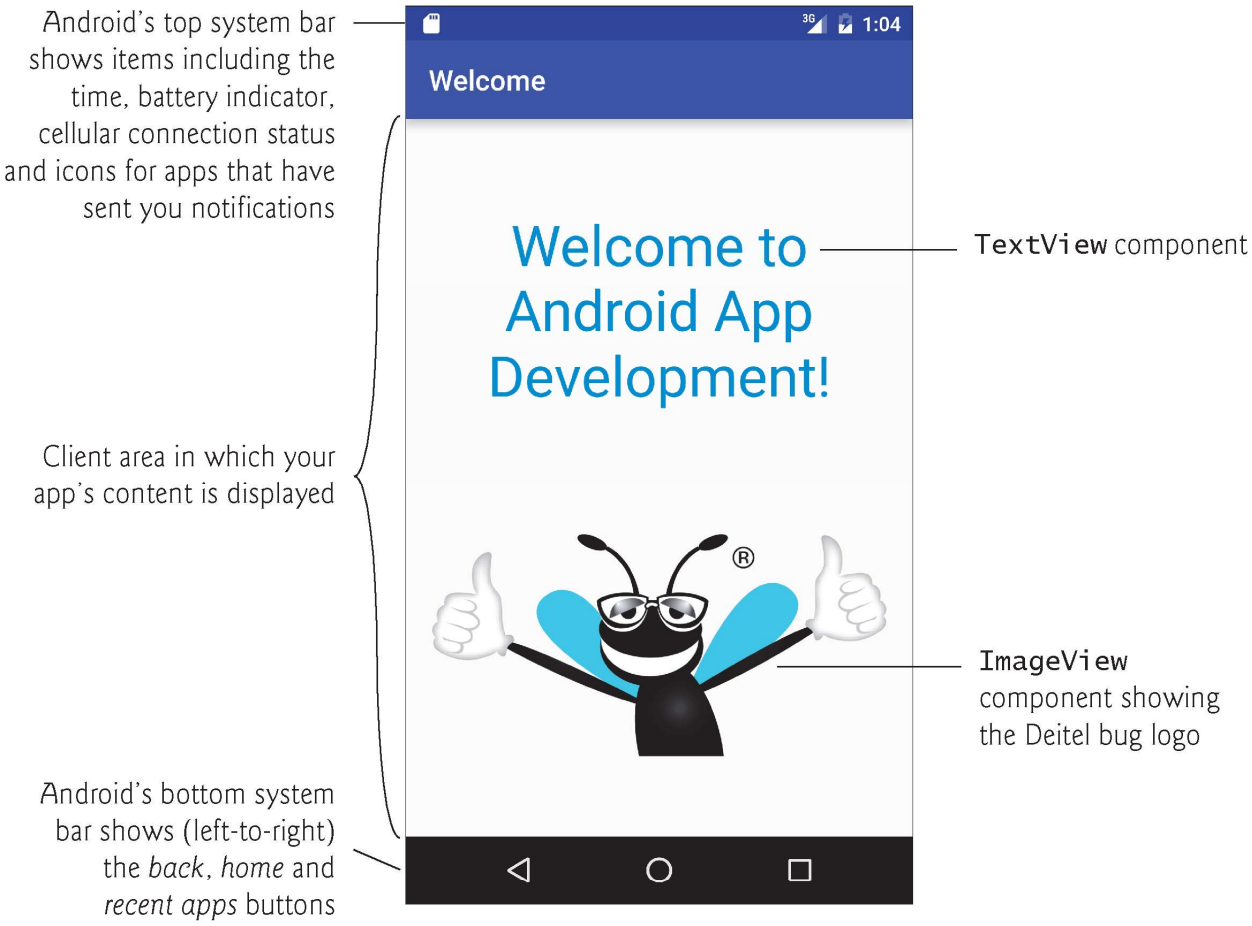


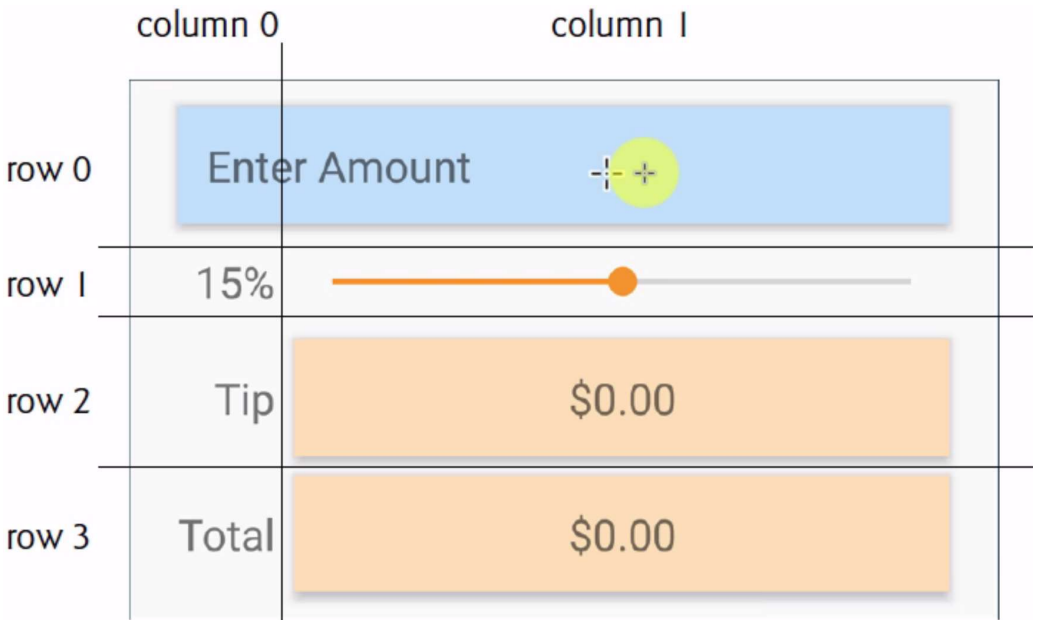
Fig. 2.1 | **Welcome** app running in the Android emulator.

Story Boarding

- As we proceed Story Boards will get more intricate, in addition to describe what *things* are, you will also have to explain what *things* do.
- For now, we will simply be sketching our designs and descriptions, but powerpoint is a great tool for Story Boarding. There are also specific applications you can download for creating story boards.
- For Example, a more complex App might be a Tip Calculator. This App has interactions that must be described…

Story Boarding with Layouts in Mind

TIP Calculator, Labeled GridLayout View, helps plan our App.



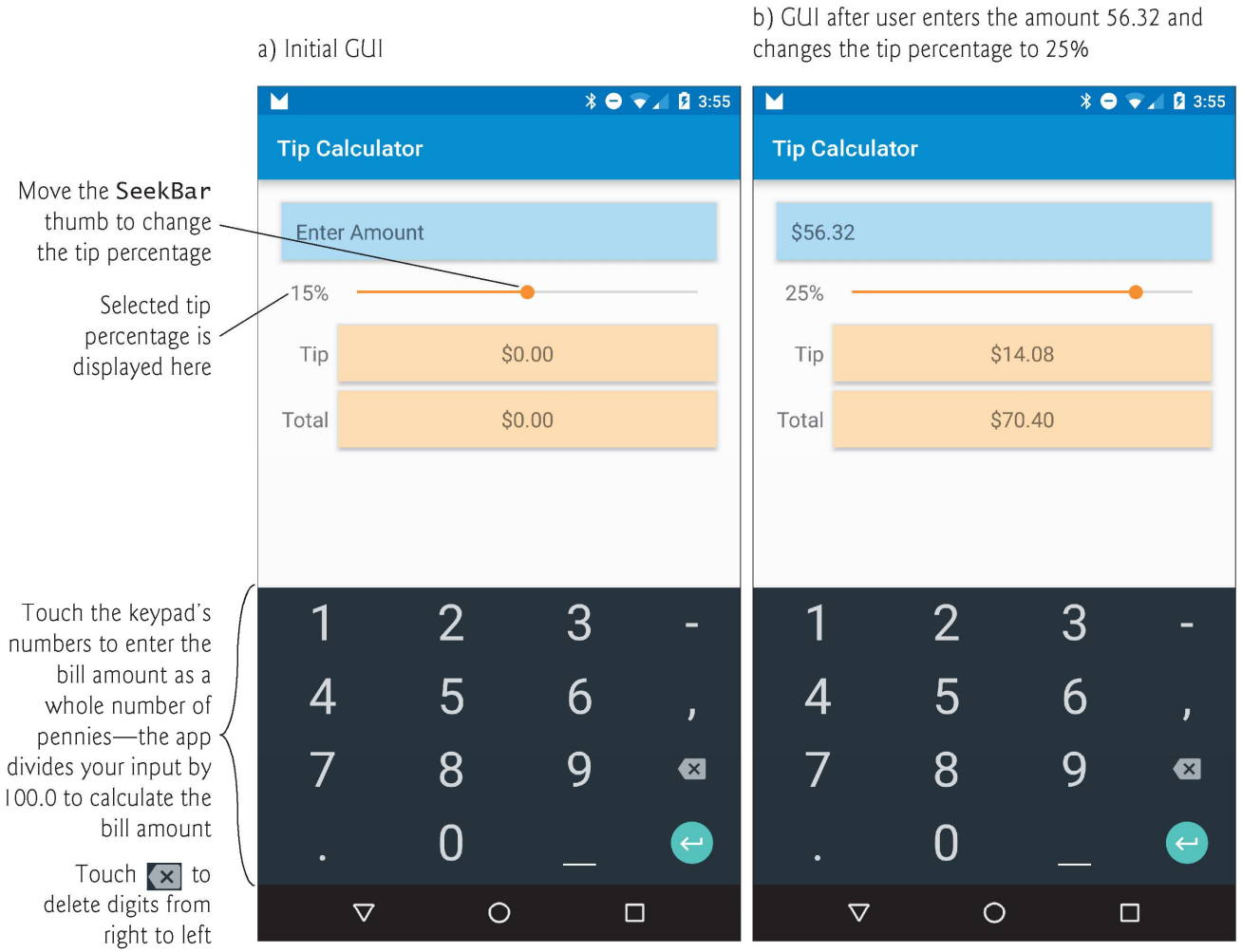


Fig. 3.1 | Entering the bill total and calculating the tip.

Brainstorming Apps

- Group work, Worksheet 1. *You Try it.*

Native EXE's, Virtual Machines & Managed Code

What is a Native Executable

- Program that runs, directly on a computer.
- *What are some IDEs/languages that can compile Native EXE's?*
  - _____
  - _____
  - _____
  - _____

- Native Executables are (typically) unbounded, can go anywhere on the computer and do anything. The OS, offers some minimal protection.
- *What are some Pros/Cons of Native EXE's?*
- Very _____, but can also be _____.

Virtual Machines can also Run Programs

- An example of this the JVM, Java Virtual Machine.
- When Java code gets compiled, it doesn't become an executable that can run directly on your computer.
  - It becomes BYTE code.
- The BYTE code is then interpreted by another running process, the JVM.
  - A VM is just a software application that emulates a physical machine.
  - The JVM is like a machine inside a machine. The running program is much more secure. *At least that was the plan when Java was invented…* ☺
- BYTE code is standard and can run on JVM's anywhere, PC's, Macs, Cars, Toasters, whatever.
- Pros/Cons?

What is Managed Code?

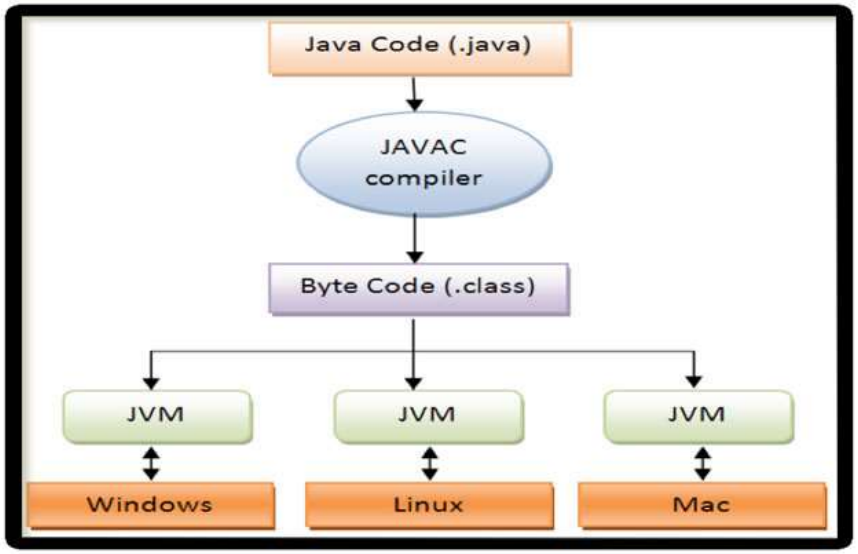
- Managed code is the next evolution of running code inside virtual machines. Came about when MS got sued by SUN (Maker of Java).
- Code is compiled into an executable. It will run directly on the CPU, but only with a Manager running with it.
- The first example of managed Code is the CLR, Common Library Runtime by Microsoft.
- C# gets compiled into a Managed EXE. The EXE will not run without the CLR.
- Not a VM. It runs “next” to the Exe, “Managing”. This includes Memory Management, Garbage Collection and Security. Eg, no Pointers in C#.

So How do Android Programs Run?

- Natively? Inside a VM? In a Managed Environment?
- All of the above are possible.
- But like Microsoft's .NET, Google's Android evolved to run in a Managed Environment, the ART, Android Runtime.
- Also like .NET, Google got sued by SUN@!!

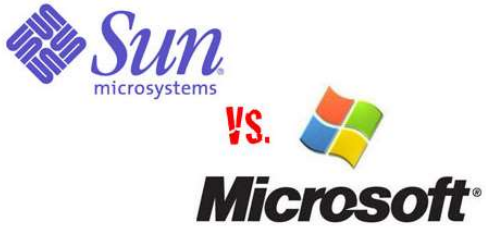
Mommy, “Where does managed code come from?”

- Java gained popularity in the late 1990’s when there were many different types of computers connecting to the Internet. HTML was X-platform, but insufficient.
- Vendors wanted a way to run their Applications on any computer with any architecture/OS. They also only wanted to write the code once. This could not be done natively, eg, with C++
  - Note: Macs were using several different chipsets (not including Intel)
- Customers wanted to be sure these Applications were secure.
- Enter Java and its JVM.



Ref: CartoonStock.com, www.slideshare.net/nextvision321/what-isja

The Battle for Managed Code



- In the late 1990's Java was getting increasing amounts of Market Share.
- Many companies had licensed Java, and integrated into their own development environments.
- Microsoft's implementation of Java would support code written on an MS platform, but wouldn't necessarily work across platforms.
- Sun sued MS for $35 Million. *(THAT'S IT ?!?)*
- They eventually settled for $20 Million, Microsoft was permanently prohibited from using "Java compatible" trademarks.

Guess Who the Big Bear is?

Ref: https://www.cnet.com/news/sun-microsoft-settle-java-suit/

The Battle for Managed Code

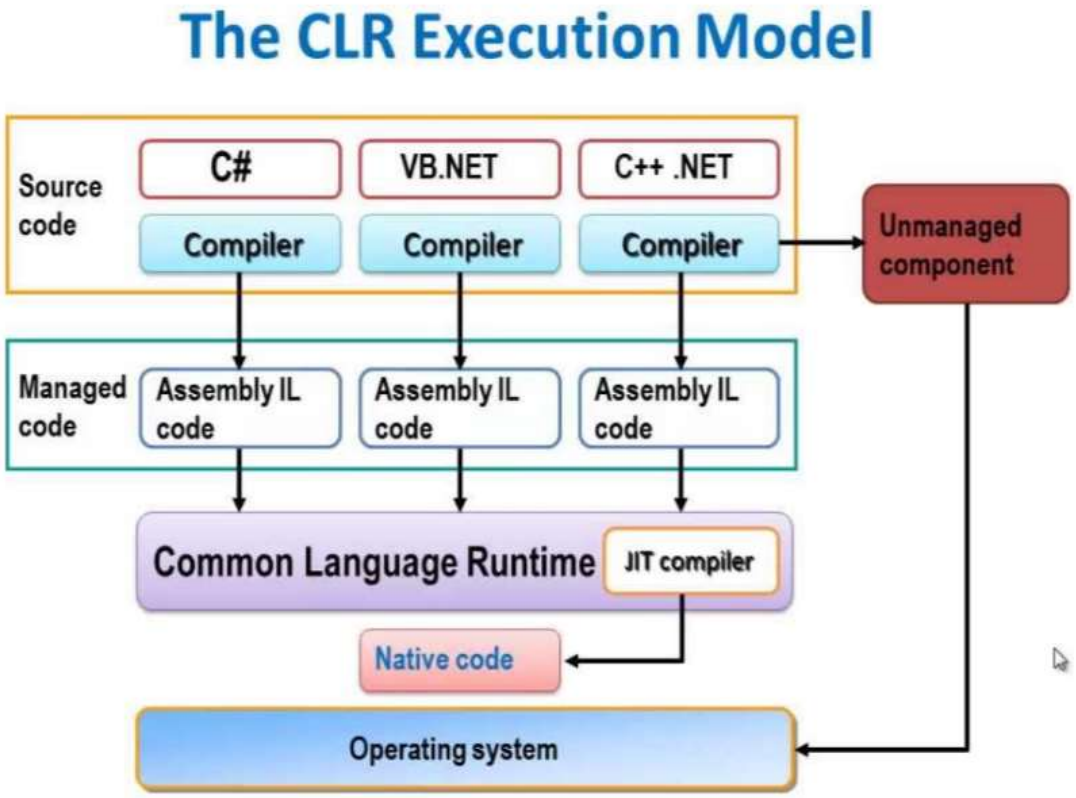
- So what did Microsoft do?
- A few years later they built their own managed code environment .NET. as well as a new programming language to go with it, C#.
- Unfortunately, Microsoft raided the best people from other companies to get this done. Eg, Borland.
- Borland lost 34 of their key people including their best Architects, including the guy who invented Turbo Pascal and Delphi, Anders Hejlsberg.
  - **Anders went on to co-invent C# at Microsoft!**

"It's like you're in the desert and Microsoft is stealing your water bottle." – Delbert Yocam (President, Borland. Sept., 1997)

- .NET is now very popular and many programming languages (eg, VB, C++, Delphi, Python, etc.) can compile down to MSIL (byte code).

How does Microsoft's Managed Code work?

- C# code compiles to an intermediate language (MSIL). Just like Java Byte Code.
- Like Java, .NET, has a virtual machine, *well sort of*, The CLR actually compiles Just in Time, then keeps the compiled code in memory for later reuse.
- You are not limited to compiling just Microsoft products. Many vendors have .NET languages.
- Notes:
  - More recent implementations of Java have a JIT.
  - The CLR is "smart". If it detects a segment of code being called frequently it will recompile with additional optimizations for better performance.



JVM vs. DVM vs. NET

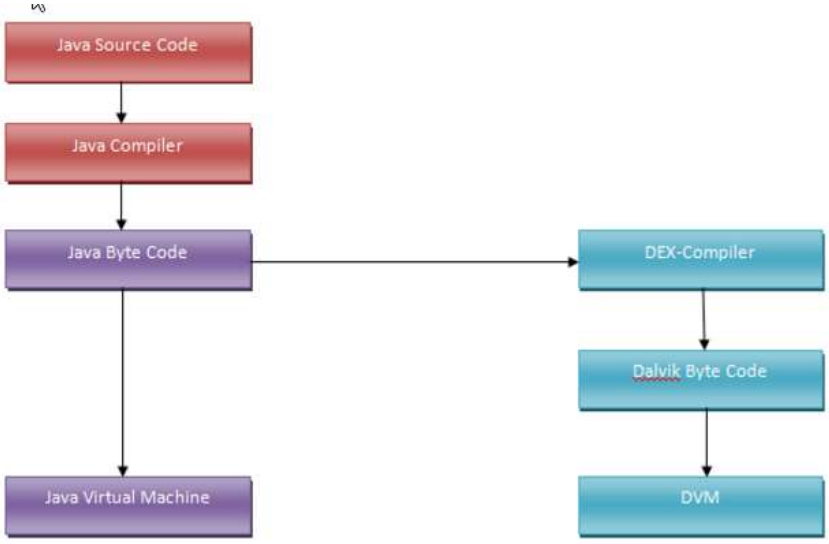
- What is the JVM again?
- DVM is very similar to the JVM, but is lightweight and can spawn itself quickly.
  - This is a must! Why?
- Every Android App runs in its own DVM.
  - It's possible to get around this, and let Apps share a DVM, but it's not typical.
  - *Well that's how it used to be anyway. Now Android uses the ART. More later…*
- DVM is very similar to JVM.
- Nice Refs:
  - http://davidehringer.com/software/android/The_Dalvik_Virtual_Machine.pdf
  - http://stackoverflow.com/questions/13577733/how-an-android-application-is-executed-on-dalvik-virtual-machine

From Source to Executable

- Native EXE's
  - Eg, C++ Source → C++ COMPILER→ NATIVE EXE
  - NATIVE EXE'S ARE FAST, BUT INSECURE.
- JVM Apps
  - EG, Java Source → Java COMPILER→ Java Byte Code →JVM
  - Slower (interpreted), more secure, runs in sandbox.
- DVM Apps are based on Java, but there is an extra compilation and a different Virtual Machine.
  - EG, Java Source → Java COMPILER→ Java Byte Code →Dalvik COMPILER → DVM
  - Takes a while to compile. Things get compiled twice!
  - Why the hassle? JVM is not Free, DVM is Free. DVM is also optimized for mobile apps?
  - Why use a VM at all? Secure, also "guaranteed" to run across different devices. Just like Java.

Compiling Android Code: Slow as a Twice Baked Potato

- Multiple Compiles
- It's still a new development environment.
- Requires Patience.
- Even the IDE's are not as feature rich.
- Let's try creating a simple app with a button.
  - We'll use a few different IDE's to compare and contrast.



- Ref: http://stackoverflow.com/questions/13577733/how-an-android-app

From Dalvik to ART

- Dalvik: Cooking a baked potato twice is delicious, but it is time consuming.
  - Why not bake the potato in advance and store in the freezer?
- **Who is ART?** *Kind of a Nerd, but with super powers.*

- Compiles code ahead of time (AOT) into runnable executable.
- Very similar to running managed code in .NET framework.
- Takes longer to install, because of the additional compilation, but runs and starts up quicker.
- Launched Experimentally in Android KitKat 4.4
- Now it's the default standard.

Physical & Virtual Device Setup

Physical Devices: Need to put your device into Dev Mode.

- Strongly Recommend physical devices.
  - Much faster!
- Typically done by clicking going to settings and clicking “About phone” or “Build Number” 7x’s.
- Then go into Developer Options and enable USB Debugging.
- Now your phone will be recognized by Android Studio.
- You may also view your phone on your screen.
- Many third party software that allow this.
- I am using MyPhoneExplorer, previously used Vysor.
  - Free version of Vysor has lots of ads.

- ***Let’s Try it.***

https://www.fosshub.com/MyPhoneExplorer.html

Virtual Device Setup

- Android Devices can be emulated.
  - AVD = Android Virtual Device
- These are typically very slow, even with tools inteneded to speed them up.
- They also consume lots of memory and can slow down your entire dev processes.
- Do not Recommend, but you may need them,
  - Eg, if you don't have a physical device.
  - Eg, to test on different versions of android.
- ***Let's Try It.***

Running Android Studio

Labels, Buttons and Checkboxes, Oh my…

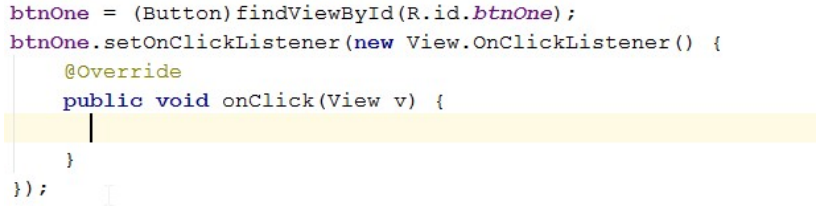
- But first, to provide some context,
  - Let's try working with a few common components in a few different languages.
- I'll Try it.
- You pick
  - C#
  - VB
  - Delphi
- Let's Try again with Android Studio.

Pinch Points: Auto Completing Event Handlers for Runtime Binding.

- Writing event handlers in Android requires too much code. How can I simplify the process?
  - Create a reference to a Button Object in Java
  - Bind the Button reference to the one instantiated in the view, by using FindViewByID(..)
  - Start typing
  - btnOne.setOnClickListener(new On…), as soon as you start typing the parameters, you will be prompted to autogenerate the wrapper. Hit Return, not Tab.

Pinch Points: Auto Completing Event Handlers for Runtime Binding.

- Be sure to click Enter, when the drop down appears.



Labels, Buttons and Checkboxes, Oh my…

- Pivoting: in other languages, eg, C#, VB, Delphi….
  - What is the equivalent of a View?
  - What is the equivalent of an Activity?
  - What is a group of Activities?

Key Concepts

- Even though you "instantiated" the Views by dragging them onto the Activity. Your Activity's Java Source doesn't have a reference.
- Don't instantiate a new object.
- Find the reference to the one that was created by Android Studio.
  - `Button btnClickMe;`
  - `btnClickMe = (Button)findViewById(R.id.btnClickMe);`
- Binding events, eg, onClick for a button is not automatic as it is with other languages. Here's the trick:
  - Start typing the event handler and let android studio fill in the mundane details.
  - `btnClickMe.setOnClickListener(new View.OnClickListener() {...}`

Key Concepts – The AVD

- Very Slow…
  - But pretty cool.
- Start your emulators early.
  - They take a while to spin up.
- If you have an android device
  - Use it, just becareful in developer mode. Your device becomes vulnerable.
  - Install Vysor – *I'll Try it.*

Let's Try It in Android Studio

-- BE PATIENT WITH THE IDE --

- Be Forewarned, Android Studio is
  - Slow at rendering code onto a design
  - Sometimes generates nonsense errors, that go away on their own.
  - Slow at running Virtual Devices
    - Load these in advance, then dynamically bind your program
  - Tricky to setup ☹
    - Disable Hyper-V in Windows
    - Ensure Virtualization is enabled in BIOS
  - Once it is setup, it should remain stable.
- Choosing a Target and Minimum SDK.
  - Why not just choose the lowest Min SDK? This would get 100% of all the devices out there.
  - Why not just choose the newest SDK as the Minimum?

Developing in Android Studio

- Very similar to developing in Java
  - Still in its infancy.
  - It's slow
  - Clunky
  - Irritating
  - But getting better.
- Be Patient! Or come back in a year or two when it's mature. ☺
  - Feel Free to complain, I will listen, but not much I can do.
  - Most irritating is probably rendering XML into a GUI
  - Painful to bind events, eg, mouse clicks.
  - Start up emulators early.
  - If you can, use a physical device to test.
  - You may occasionally need to restart Android Studio and/or emulator
  - Don't wait til last minute on assignments.

GUI Development

- GUI Development is a bit different than other types of programming.
  - How many have written Desktop GUI Applications?
- The core programming concepts are the same:
  - Variables/Data Types, Abstraction, Inheritance, Polymorphism, etc..
- What's Different?
  - Components
  - Layouts
  - Events

GUI Development and Event Based Design

- In Java, developing GUI's can be done Manually or using a graphical IDE like WindowBuilder.
- What's the difference?
  - Manual development: Must describe in code the layout and handling of events.
  - Automatic (WindowBuilder):
    - You can drag and drop components onto a form.
    - You can also add event handler wrappers. Very Nice!
  - Neither approach allows you to easily create events to go along with your components. ☹
  - In C#, Visual Basic, Delphi, and other high level languages, both layout and event handling are handled automatically.
- In android development studio (which is really just a Java IDE in disguise) you can use the graphical interface.
- Demo other IDE's, you pick one and I will build a hello world GUI. *Let's Try It.*

Preparation for Next Week

Preparation for Next Week

- Be ready to do group work for an actual application.
  - Install and configure Android Studio on your laptop and bring it.
  - Setup Emulator and run hello world.
  - *There are many types of computers with many types of configurations. I can offer suggestions to help if you get stuck installing Android Studio, but cannot performed detailed troubleshooting.*

Closing Info

- Syllabus & Homework will be posted on Piazza tomorrow.
- Be sure you have the textbook, you will need it for the HW.
- If you are not on Piazza, eg, registered late, email me ASAP. sse@bu.edu