CS 646 Android Mobile Application Development
Spring Semester, 2015
Syllabus

San Diego State University -- This page last updated 1/20/15

CS 646 Syllabus

Instructor	Roger Whitney
Office	GMCS 561
Phone	594-3535
Email	whitney@sdsu.edu
Office Hours	3:15-5:15 pm Tuesday,Thursday, 10 am - Noon Friday

Course Objectives: Learn the basics of developing Android apps.

Course WWW Site: The course website is in the SDSU Blackboard system. SDSU students can access it through their SDSU portal. It can also be access at: https://blackboard.sdsu.edu/

Class Location and Time: The class meets in room GC 1504 (Gateway Center) from 7:00 pm to 8:15 pm Tuesdays and Thursdays.

Textbook:

- Android Programming: The Big Nerd Ranch Guide, Phillips & Hardy, Big Nerd Ranch Guides, April 7, 2013, ISBN 978-0321804334

Prerequisites: The course assumes that you know Java. All assignments in the course will use the Java programming language to develop Android apps. This involves designing and building GUIs and using SQL databases. While prior experience with GUIs and SQL databases are not required prior experience will be very useful in the course.

Required Hardware: You can use a Windows, Mac or Linux based computer to develop Android apps. While it is not strictly required to have an Android device it is strong recommended that you have access to one. There is an Android emulator but it is very slow and there are a number of things the emulator can not do. The department does have a few Android phones for SDSU students to use in this course.

On-Line Students: All the course materials are placed on-line via blackboard for access by all students. The class lectures are recorded and can be accessed in realtime during class and after the class is over. During the class students not in classroom can ask questions via skype. The class skype name is cs646.sdsu. There is an on-line discussion board for questions outside of class. On-line students in the San Diego area are welcome to attend class.

Grading: Your grade will be based on one exam (150 points) assignments (650 points), a project (200 points). There is no extra credit work in this course. The course will be graded on

1,000 points. There will be 5 assignments. All assignments are worth 130 points. Tentative due dates for the assignments are Feb 1, Feb 15, March 1, March 22, and April 9. The exam will be March 10.

Remote Students and Exams: Off campus students need to find someone to proctor them while they take the March 10 exam. I need to know who your proctor will be by Feb 19 so I can get the exam to them in time.

Qualcomm Donation: Thanks to a generous donation from Qualcomm we will have a few Android Phones to use during this course.

Course Project: There is a course project required for this course. You can work on the the course project by your self or in teams of two people. Teams larger than 2 are not allowed. You are to come up with an idea for an Android application and implement the application. The project is due May 14. Your project will be evaluated using the following criteria:

Originality (5 points)
: How original is the idea and/or implementation of the project.

Size of project (25 points)
: Is the project an appropriate size for a semester project. A project that is too large is as bad as a project that is too small. The project should be larger in scope than an assignments in the course? The project should take longer than 2-3 week to design and implement?

Quality of UI (100 points)
: How well does the app follow users mental model of how the app should work. Are the UI elements used effectively or not? Is the app UI structured in a way to make it easy to use and understand. Is the text used in menus, labels, buttons, etc. concise and have clear meaning? Is the app complete, that is does it correctly handle the lifecycle of its activities, does it support a range of screen sizes and device rotation, does it support user preferences, etc.

Working code (78 points)
: Do the features implemented work. All UI elements should actually do what they are supposed to do? Are features fully or only partially implemented. Does the app have enough features to actually do something. Does the app run? Are there bugs and memory leaks? Does the app crash?

Quality of code (22 points)
: The code should be formatted in a reasonable and consistent manner. Names of classes, methods and variables should understandable and follow standard naming conventions. The code should be clear and well organized. The code should be appropriately documented.

Crash Policy: The last day to add this course is February 4.

Dropping the course. If you drop this course CES will refund part of your fees depending on when you drop the course. If you drop the course at least one day before the first time the class meets you will receive a full refund minus $17 administration fee. If you drop the course after the first meeting of the class but before 25% of the course time as elapsed you will receive 65% of your fees back minus $17 administration fee. After this date you will not receive refund if you drop the course. If you wish to drop this course to take another of the

Certificate course this semester you can do so without any penalty. If you have questions about this and other issues related to CES please contact the CES registration office at 619-594-5152.

Late Policy: Late homework will be accepted, but with a penalty. An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eight day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.

Cheating: Any one caught cheating will fail the course and they will be reported to the SDSU Judicial Procedures Office.

My Dad:
Over winter break my father was diagnosed with cancer. The cancer has spread and cannot be treated. I will be spending part of the semester helping taking care of him and will be out of town. During that time the course will be offered on-line. I will be recording lectures and posting them on-line. Details of this will be discussed when the class meets.

Course Goals and Outcomes:
- Design Android apps using Java.
- Use Android IDEs and API effectively in Android applications.
- Implement and debug Android apps.
- Separate resources (text, sound, images) from Android code to facilitate working with graphic designers and localization of apps.
- Measure resource consumption in Android applications to help optimize performance.
- Support multiple versions of Android OS and multiple Android devices with one app.
- Develop Android GUI interfaces separate from Amdroid app code.

This will improve your ability to:
- Analyze a problem, and identify and define the computing requirements appropriate to its solution
- Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- Use current techniques, skills, and tools necessary for computing practice.

Topics covered in the course:

- Using Eclipse/Android Studio
- Android Activities and Activity life cycle
- Fragments
- Layouts, Intents & Intent Filters
- Permissions
- GUI widgets
    - Labels, buttons, Check boxes, text fields, Lists, pickers, etc.
- Containers and Keyboards
- Layout styles
- Threads and Concurrency
- Fonts
- WebViews
- Services
- AppWidgets
- Files, Preferences, SQLite database
- Notifications, Resources
- Network Access
    - JSON, SOAP
- 2D Graphics
- Maps
- Dealing with multiple screen sizes
- Testing
- Tablets
- Sensors
    - Touch, gyroscope, accelerometers

Disabled Students: If you are a student with a disability and believe you will need accommodations for this class, it is your responsibility to contact Student Disability Services at (619) 594-6473. To avoid any delay in the receipt of your accommodations, you should contact Student Disability Services as soon as possible. Please note that accommodations are not retroactive, and that accommodations based upon disability cannot be provided until you have presented your instructor with an accommodation letter from Student Disability Services. Your cooperation is appreciated.

CS 646 Android Mobile Application Development
Spring Semester, 2015
Assignment 1

San Diego State University -- This page last updated 1/23/15

Due: Feb 1 at 23:59
Points: 130

Objectives

Learn how to use Eclipse or Android Studio to create an Android app
Handle basic Activity lifecycle
Use basic GUI widgets
Handle Screen rotation

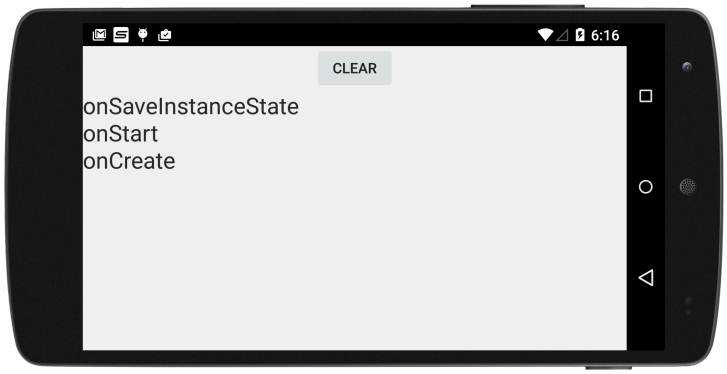
Life Cycle Events

You are going to create an app that logs the standard life cycle activity methods (onCreate, onRestart, onStart, onPause, onSaveInstanceState, onRestoreInstanceState and onResume) are called. Create an app with one activity and implement the methods onCreate, onRestart, onStart, on Pause, onSaveInstanceState, onRestoreInstanceState and onResume in the activity. Look up the methods in the documentation for their signatures. (Which ones require you call super?) Each time one of the method () is called you are to do two things:

1 Write the name of the method to the log (LogCat)

2 Append the name of the method as a new line of text at the end of a TextView in the interface of the app.

Create two layouts for the activity, one landscape and one portrait. In each layout you need a TextView and a button with the label “Clear”. The TextView displays the methods that have been called in the app, one method per line. When the “Clear” button is pressed the text in the TextView is cleared. Note the change of location of the button between landscape and portrait view.

Landscape View	Portrait View
	

Each time one of the life cycle methods is called the name of the method should be added to the TextView and a statement added to the log indicating which method was called. When the device or emulator changes orientation the corresponding layout is to be used.

When you create your project set the minimum required SDK to API 15.

Issues

1. When you run your app and rotate the device/emulator are the method displayed in the TextView consistent with methods called in the log? If not what would you have to do to make them consistent?

Grading

The assignment will be graded as follows:

Points	Item
30	Two layouts, Each used in proper orientation
30	Clear Button works in both layouts
30	Correct values for methods displayed when rotate device
15	Logging methods
15	Non numeric text displayed on screen defined in string resource
5	Coding style

Points	Item
5	Answers to issue 1.

What to turn in

Add to your project a file called "ReadMe.txt". In this file add the answers to the one issue. Create a zip file of your entire android project. This can be done using Eclipse. Select the "Export …" option from "File" menu, then select "Archive File" option under "General". Please no rar files. If you are using Android Studio you need to use another program to zip your project directory. Turn in your assignment at: http://bismarck.sdsu.edu/CoursePortal. There is a link to that site in the assignment section of the course Blackboard site. You will need to create a password for the assignment site the first time you use it. Once you have logged on to the site select the assignments tab.

CS 646 Android Mobile Application Development
Spring Semester, 2014
Assignment 2

San Diego State University -- This page last updated 2/3/15

Due: Feb 15 at 23:59
Points: 130
Objectives

Use some basic UI widgets
Use Intents to call other Activities
Handling the Keyboard
Using Fragments

UI Sampler

You will be trying out a number of different UI elements. The combination of elements does not make much sense as an app, but does give you experience with more UI elements. You do not need different layouts for landscape and portrait views.

This assignment will have a number of different activities. The first (main) activity contains a spinner, a button, a list fragment (the same list fragment defined in the List Activity) and an EditText field. The spinner contains a list of activities one can select. When the user selects the option and then clicks on the button the selected activity becomes the active activity. The contents of the text field are sent to the next activity. Use descriptive names in the spinner for your activities. Class names are not meaningful to app users. Each section below describes an activity that the user can go to.

Date Activity

The date activity shows a date picker. When the user selects a date, display a dialog confirming the choice. If user confirms the choice then save the date in permanent storage. When the user restarts the app the date picker should display the last selected date. When the user goes back to the first (main) activity the date should show it the EditText field.

Keyboard Activity

This activity's view has three EditText fields. One at the top of the screen, one in the middle of the screen and one at the bottom of the screen. The top EditText field should contain the text sent from the first (main) activity. There is also two buttons. One labeled "Hide" and the other labeled "Back". The "Back" button when tapped goes back to the first (main) activity. Note creating a new version of the first activity and going to it is not the same as going back to the existing activity. When the user taps on one of the EditText fields the soft keyboard appears. When the user taps on the "Hide" button the soft keyboard disappears. One problem with the soft keyboard is that it can cover up part of the screen. In this case it will hide the field on the bottom of the screen. You activity should have the view pan up when the soft keyboard is shown and the focus is in the bottom EditText field.

List Activity

This activity has a list fragment and a back button. The list contains the of deserts given below. The same list fragment appears in the first (main) activity. The user can select one item in the list. When the user taps on the back button the app goes back to the first (main) activity. The desert selected in the List Activity should be selected in the first activity. If the user selects a desert on the first (main) activity then it should be selected when the user goes to the List Activity.

Cupcake
Donut
Gingerbread
Ice Cream
Jelly Bean

Extra Credit

1. Enable the application icon in the action bar to take the user back to the first (main) activity. Make sure that the app goes back to the first activity and not create a new instance of the first activity. If the app is displaying a one of the other activities, that activity needs to be destroyed when you go back to the first activity.

2. Add items to the Action Bar to allow the user to select which activity to go to.

3. Enable the split action bar.

4. Add a context menus for deserts.

Grading

The assignment will be graded as follows:

Points	Item
30	**Main View (Spinner, Can go to next activity, Pass Data to next activity)**
30	**Keyboard Activity works correctly (Panning, back, hide, field contains data from first activity)**
30	**Date Activity**
10	**Saving date data permanently and retrieving the data**
30	**List Activity works correctly (List items correctly and send selected item back to main activity, uses Fragments)**
5	**Back in Action Bar (Extra Credit)**
5	**Select Activities in Action Bar (Extra Credit)**
5	**Split Action Bar (Extra Credit)**
5	**Context menus (Extra Credit)**

What to turn in

Zip your Eclipse project and upload it to the course portal. Turn in your assignment at: http://bismarck.sdsu.edu/CoursePortal. There is a link to that site in the assignment section of the course Blackboard site. You will need to create a password for the assignment site the first time you use it. Once you have logged on to the site select the assignments tab.