Colorado State University GLOBAL CAMPUS

CSC475: PLATFORM-BASED DEVELOPMENT

Credit Hours: 3

Contact Hours: This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course's reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

Faculty Information: Faculty contact information and office hours can be found on the faculty profile page.

Course Description and Outcomes

Course Description:

In this course, students will demonstrate an understanding of programming applications in a mobile application development environment. Students will be introduced to programming constructs associated with mobile application development including building graphical user interfaces, saving data for application use, and mobile application deployment.

Course Overview:

In this course, students will be introduced to building professional mobile applications using Java and the Android SDK. Students will perfect the skills needed to create reliable software using Android activities, services, and intents. These skills will be broad enough to allow students to develop user interfaces that work with a variety of mobile phones and tablets. Students will be introduced to programming techniques needed to store and retrieve data in files, databases, and content providers. Students will also be exposed to integrating applications with web services and to publishing secure Android applications. Students will utilize their previous programming knowledge to develop basic mobile applications, build a graphical user interface, and save, execute, debug, and deploy mobile applications.

Course Learning Outcomes:

1. Develop mobile applications using the Android IDE and Gradle.
2. Understand how to develop and manipulate Android activities, services, and intents.
3. Create cross-platform interactive and responsive user interfaces.
4. Demonstrate how to save data for persistent use within a mobile application by using files, databases, and content providers.
5. Integrate Android applications with web services.
6. Deploy a mobile application for usage on a mobile environment.

Participation & Attendance

Prompt and consistent attendance in your online courses is essential for your success at CSU-Global Campus. Failure to verify your attendance within the first seven days of this course may result in your withdrawal. If for some reason you would like to drop a course, please contact your advisor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

Course Materials

Required:
Deitel, P. J., & Deitel, H. M. (2017). *Android: How to program* (3rd ed.). Upper Saddle River, NJ: Pearson. ISBN-13: 978-0134444307

NOTE: *All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Please read through each course module carefully.*

Course Schedule

Due Dates
The Academic Week at CSU-Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT, and peer responses posted by Sunday 11:59 p.m. MT. Late posts may not be awarded points.
- **Opening Exercises:** Take the Opening Exercise before reading each week's content to see which areas you will need to focus on. You may take these exercises as many times as you want. The Opening Exercises will not affect your final grade.
- **Mastery Exercises:** Students may access and retake Mastery Exercises through the last day of class until they achieve the scores they desire.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

Weekly Reading and Assignment Details

Module 1

Readings

- Chapter 1 in *Android: How to Program*

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (20 points)

Execute an Android Studio Application

For this first Critical Thinking assignment, you will locate and execute an Android Studio application. Each chapter in the course textbook has its own set of data files, which can be download from http://www.deitel.com/books/AndroidHTP3/ (download code examples for data files and save them to your computer).

Below are the steps for completing this assignment:

- First, you will need to install Android Studio and the Java Development Kit (JDK) using the steps outlined in the book "Before You Begin" section. Follow the instructions on **Page xxxvii: Before You Begin** in the course text.
- Launch Android Studio. When the application is opened, take a screen shot, showing your computer desktop on the background to ensure that you completed this activity in your computer.
- Locate and then open the androidhtp3_examples\AndroidProjects folder in the downloaded data files on your computer's hard disk. You will need to import the **TipCalulator app.** Take a screen shot of the application, showing your computer desktop on the background to ensure that you completed this activity in your computer.

Assignment deliverables (all in a single Word or PDF file, with steps numbered, and steps in order):

1. A screenshot of Android Studio showing it was successfully installed on your computer.
2. A screenshot of the **Tip Calculator** program executing in the Android Virtual Device (refer to Chapter 1 section 1.9 in your textbook).

Ensure that your assignment deliverables are attached before submitting your assignmentAssignment Description Here

MODULE 2

Readings

- Chapter 2 in *Android: How to Program*
- Android Developers. (n.d.). Application Fundamentals. Retrieved from https://developer.android.com/guide/components/fundamentals.html

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

Creating a Scrapbooking App Application

For this second Critical Thinking assignment, you will use the Welcome application in the textbook source code as a starting point.

1. Find three open source images of famous landmarks using websites such as Flickr or Pixabay.
2. Create an app in which you arrange the images in a collage.
3. Add text that identifies each landmark.
4. Recall that image file names must use all lowercase letters.

Assignment deliverables (all in a ZIP file):

1. A screenshot showing the execution of your application.
2. A detailed overview of the steps taken to complete your application.
3. The project folder containing all the source code for your application.

Ensure that all of your assignment deliverables are attached before submitting your assignment

MODULE 3

Readings

- Chapter 3 in *Android: How to Program*
- Android Developers. (n.d.). Layouts. Retrieved from https://developer.android.com/guide/topics/ui/declaring-layout.html

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

Creating a Mortgage Calculator App Application

For this third Critical Thinking assignment, you will create a mortgage calculator app that allows the user to enter a purchase price, down-payment amount, and an interest rate.

1. Based on these values, the app should calculate the loan amount (purchase price minus down payment) and display the monthly payment for 10, 20, and 30-year loans.
2. Allow the user to select a custom loan duration (in years) by using a SeekBar and display the monthly payment for that custom loan duration.

Assignment deliverables (all in a ZIP file):

1. A screenshot showing the execution of your application.
2. A detailed overview of the steps taken to complete your application.
3. The Project folder containing *all* source code for your application.

Ensure that your assignment deliverables are attached before submitting your assignment.

MODULE 4

Readings

- Chapter 4 in *Android: How to Program*
- Android Developers. (n.d.). Fragments. Retrieved from https://developer.android.com/guide/components/fragments.html

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

Creating a Road Sign Test App Application

For this fourth Critical Thinking assignment, you will create an app that tests the user's knowledge of road signs. Display a random sign image and ask the user to select the sign's name. Visit http://mutcd.fhwa.dot.gov/ser-shs_millennium.htm for traffic-sign images and information.

Assignment deliverables (all in a ZIP file):

1. A screenshot showing the execution of your application.
2. A detailed overview of the steps taken to complete your application.
3. The project folder containing all source code for your application.

Ensure that your assignment deliverables are attached before submitting your assignment.

MODULE 5

Readings

- Android Developers. (n.d.). App data and files. Retrieved from https://developer.android.com/guide/topics/data/index.html

- Chugh, A. (2018, April 2). Android internal storage example tutorial. [Blog post]. *JournalDev.* Retrieved from https://www.journaldev.com/9383/android-internal-storage-example-tutorial?utm_source=website&utm_medium=sidebar&utm_campaign=Android-Sidebar-Widget
- Chugh, A. (2018, April 2). Android shared preferences example tutorial. [Blog post]. *JournalDev*. Retrieved from https://www.journaldev.com/9412/android-shared-preferences-example-tutorial?utm_source=website&utm_medium=sidebar&utm_campaign=Android-Sidebar-Widget
- Chugh, A. (2018, April 2). Android SQLite database example tutorial. [Blog post]. *JournalDev.* Retrieved from https://www.journaldev.com/9438/android-sqlite-database-example-tutorial?utm_source=website&utm_medium=sidebar&utm_campaign=Android-Sidebar-Widget

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

Creating a Data Access Object with SQLite

For this fifth Critical Thinking assignment you will modify an existing app that tracks business expenses. Download the existing app from <download site> and explore the application. There are two inner classes: ExpenseItem, which defines constants used in mapping the Expense data to the database, and Helper. You should see that the Expense class is a simple POJO with properties of amount, description, and expenseDate.

A DAO is an excellent and widely-used design pattern where the code to access a data source is encapsulated within a single class.

1. Open edu.csuglobal.expenses.data.DAO.java.
2. Inside the DAO constructor near line 35, add the code to initialize an instance of the database helper class at the location marked TODO.
3. Locate the queryExpenses() method and complete the steps marked TODO.
4. Expand app | java |edu.csug.expenses(androidTest) | data.
5. Right-click DAOTest and select Run 'DAOTest'. If you are offered a choice of two DAOTest options, take the first one, which has the Android symbol next to it. When prompted to choose a device, select the Galaxy-Nexus AVD and click OK.
6. Click the “4:Run” button at the bottom of the Android Studio window to show the test results.
7. Open edu.csug.expenses.ExpensesListActivity.java. In the following steps, you may see class names marked as errors due to the required imports not being in the code. The simple solution is to click the class name. When prompted, press <Alt><Enter>. Android Studio will add the required import.
8. Locate the onCreate() method and complete the steps marked TODO.
9. If it is not already running, start the Galaxy-Nexus AVD. Make sure the app is selected next to the Run button, then run the application by clicking the green run arrow as normal. You should see the list of sample expenses.
10. At the end of the onCreate() method in ExpensesListActivity, add a call to registerForContextMenu(). Use “mList” as the reference for the View; this should display the context menu.
11. Override onCreateContextMenu() and inflate the menu with an id of R.menu.context_menu. Right-clicking on the file and selecting Generate | Override Methods... is the easy way to create the new method.
12. Save and execute the application.
13. Click and hold on one of the expenses in the list. Your context menu should appear. Of course, there is no associated functionality yet.
14. Add a new method public int deleteExpensesById(int id) to the DAO.

15. Create a where clause, get an instance of a writable database, then call the delete method.
16. Uncomment the onContextItemSelected() code at the bottom of ExpensesListActivity. Organize imports as needed.
17. Test your work.

Assignment deliverables (all in a ZIP file):

1. A screenshot showing the execution of your application.
2. A detailed overview of the steps taken to complete your application.
3. The project folder containing all source code for your application.

Ensure that your assignment deliverables are attached before submitting your assignment.

MODULE 6

Readings

- Chapter 7 in *Android: How to Program*

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Critical Thinking (70 points)

In this exercise, you will:

- Modify an existing Android client application to send a service request to a java servlet. The java servlet (server) will reply back to the Android client with a response.
- Configure and run a java servlet (server) to run on your local machine.

1. Configure the Java servlet.
   Download the latest Eclipse IDE (Oxygen) and create a new dynamic web project. Use the Java source file supplied by your instructor as the source code for your dynamic web project. Explore the code and note the implementations of the get and post methods. Launch the java servlet by using the built-in Tomcat Server that comes with the latest Eclipse Oxygen distribution.
2. Take a screenshot of the Java servlet running. Note that the Eclipse Oxygen environment has a built-in web browser. In your screenshot, add a textbox and in the textbox, identify the service that is being performed by the servlet.
3. Obtain the Android client app form your instructor. The name of this client app is "HttpExample." This example already has the built-in code to send requests and receive responses from an external server. Take screenshots of the messages being displayed after clicking on the "GET METHOD" and "POST METHOD" buttons.
4. In this step, you will modify your Android client to "consume" the service provided by the Java servlet running on your local host machine. Explore the code the Android app. Note how the button click events are used to send a request and receive a response. Using this code pattern, add a new button and a data entry textbox to send user data (from the data entry textbox) to the Java servlet when the button is clicked. Write the Java servlet response into the original Android app's textbox.

Take screenshots of your Android app before requesting the service and after receiving a response.

Assignment Deliverables (all in a ZIP file):

1. The screenshots list in the specifications above.
2. A detailed overview of the steps taken to complete your application.
3. The project folder containing all source code for your application.

Ensure that your assignment deliverables are attached before submitting your assignment.

MODULE 7

Readings

- Google Developers. (2018, March 26). Get started. Retrieved from https://developers.google.com/maps/documentation/android-api/start
- Google Developers. (2018, April 13). Controls and gestures. Retrieved from https://developers.google.com/maps/documentation/android-api/controls
- Google Developers. (2018, April 13). Map objects. Retrieved from https://developers.google.com/maps/documentation/android-api/map

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

MODULE 8

Readings

- Chapter 10 in *Android: How to Program*

Opening Exercise (0 points)

Discussion (25 points)

Mastery Exercise (10 points)

Portfolio Project (350 points)
Immersive Android Mobile Application

Throughout the course, you have been exposed to a number of technologies that can be utilized in order to create an immersive Android mobile application. For your Portfolio Project, you will identify a prospective idea for development as an Android mobile application. You will create the interface for the application and identify features for the application. You do not need to full implement the application, but instead you should provide a 4-5 page (not including title and reference pages) analysis of the required technologies that would need to be included for your Android Application.

Your paper must be formatted according to the CSU-Global Guide to Writing and APA and be supported by a minimum of three professional and academic sources. The CSU-Global library is a good place to locate these sources. Review the rubric in the Module 8 Materials folder for specific grading criteria.

COURSE POLICIES

Course Grading

20% Discussion Participation
0% Opening Exercises
8% Mastery Exercises
37% Critical Thinking Assignments
35% Final Portfolio Project

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

IN-CLASSROOM POLICIES

For information on late work and incomplete grade policies, please refer to our **In-Classroom Student Policies and Guidelines** or the Academic Catalog for comprehensive documentation of CSU-Global institutional policies.

Academic Integrity
Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /repurposing your own work (see *CSU-Global Guide to Writing and APA Requirements* for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU-Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the Library and Internet resources.

Citing Sources with APA Style
All students are expected to follow the *CSU-Global Guide to Writing and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on CSU-Global APA style, please review the APA resources within the CSU-Global Library under the "APA Guide & Resources" link. A link to this document should also be provided within most assignment descriptions in your course.

Disability Services Statement
CSU-Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email ada@CSUGlobal.edu for additional information to coordinate reasonable accommodations for students with documented disabilities.

Netiquette
Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults, or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments of an ethnic, religious, age, disability, sexist (or sexual orientation), or racist nature; and the unwanted sexual advances or intimidations by email, or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, please let your instructor know.