Android: Student's Choice

due by noon ET on Wed 3/21

Ingredients.

- Android SDK
- Creativity
- Java
- …

Help.

Help is available throughout the week at `http://help.cs76.net/`! We'll do our best to respond within 24 hours. Be sure, though, to take advantage of lectures and sections as well as videos thereof!

Academic Honesty

All work that you do toward fulfillment of this course's expectations must be your own unless collaboration is explicitly allowed by some project. Viewing, requesting, or copying another individual's work or lifting material from a book, magazine, website, or other source—even in part—and presenting it as your own constitutes academic dishonesty, as does showing or giving your work, even in part, to another student.

Similarly is dual submission academic dishonesty: you may not submit the same or similar work to this course that you have submitted or will submit to another. Nor may you provide or make available your or other students' solutions to projects to individuals who take or may take this course (or CSCI S-76) in the future.

You are welcome to discuss the course's material with others in order to better understand it. You may even discuss problem sets with classmates, but you may not share code. You may also turn to the Web for instruction beyond the course's lectures and sections, for references, and for solutions to technical difficulties, but not for outright solutions to problems on projects. However, failure to cite (as with comments) the origin of any code or technique that you do discover outside of the course's lectures and sections (even while respecting these constraints) and then integrate into your own work may be considered academic dishonesty.

If in doubt as to the appropriateness of some discussion or action, contact the staff.

All forms of academic dishonesty are dealt with harshly.

Grades.

Your work on this project will be evaluated along four primary axes.

Correctness. To what extent is your code consistent with our specifications and free of bugs?
Design. To what extent is your code written well (*i.e.*, clearly, efficiently, elegantly, and/or logically)?
Scope. To what extent does your code implement the features required by our specification?
Style. To what extent is your code readable (*i.e.*, commented and indented with variables aptly named)?

Student's Choice.

☐ Your task, by the project's deadline, is to implement the project whose proposal was approved by your TF! In addition to the features you have required of yourself, we do have some additional specifications.

The course will conclude with the App Party, a course-wide exhibition of the students' choice projects (like this one!) and will be an opportunity to mingle with classes, see each other's work, and see cake.

Requirements.

☐ By now, you should've heard back from your TF either approving your project or requesting changes to the proposal. You should not proceed with your project's implementation until it has been approved. If it's not yet approved, do feel free to drop your TF a note to inquire its status. Reasonable changes to your app are allowed with approval from your TF.

☐ Along with your project, you will need to provide some documentation so we know how your project works and what it should do. Once your project is complete, write a document that describes your apps usage, its purpose, and any setup that's required to get it working properly. In essence, reading through this documentation should provide us with enough information so that we understand your app and the entirety of its usage without requiring us to ask you any questions.

☐ In addition to your source code and documentation you will also provide a pre-compiled APK file that serves as a final build of your app; something we can install in an emulator or device that meets the minimum requirements of your app and run without having to compile your Eclipse project. Be sure to read through the **How to Submit** directions, below, so that you properly submit all required files.

Implementation Details.

☐ The application should have a minimum SDK API level of 7. You may use a later version If and only if some feature important to your project requires it. If this is the case, you may modify your project to use a later API version but you must mark in your source code, using Java comments, the precise line that requires the API version you are using. Be sure the text **API CHANGE** appears in your comment so that we can easily find it via a search.

☐ Under no circumstances should we be able to cause your program to crash at runtime.

☐ You may select your own project and application name.

☐ Your project's package name should be: `net.cs76.projects.student########`, where `########` is your 8-digit Harvard ID (HUID), the same credential that you use to log into `help.cs76.net`.

☐ You must include your name, email, and HUID in a comment atop every Java source file in the project.

How to Submit.

☐ Before the project's due date, export your project in Eclipse for submission. Open Eclipse and click the **File** menu and then **Export**. In the window that appears, click on the triangle next to the **General** section so that you can see the options contained within. Select **Archive File** and click the **Next** button. On the next window, check the box directly to the left of your project you'd like to submit. Be sure no others are selected, or you will export those as well. Click on the **Browse** button to select a location you'd like to export the ZIP file and be sure to name it `########.zip`, where `########` is your 8-digit Harvard ID (HUID), the same credential that you use to log into `help.cs76.net`.

After selecting where the ZIP file will be placed, make sure the export options are correct. Notably that you are saving as a ZIP file (and not tar), that a directory structure is created for files, and that the contents are compressed. When ready, click **Finish** to export your app.

☐ Create a separate ZIP file called `########-app.zip` that contains the following:
  ☐ `Readme.pdf` – your project's documentation. This file may also be an HTML or plaintext file, if you prefer, and called `Readme.html` or `Readme.txt`, respectively. This file must include your name, email, and HUID at the top of every page.
  ☐ A pre-compiled APK file. You'll need to export your APK from Eclipse by going to the **File** menu, **Export**, then selecting **Export Android Application** from the **Android** section. Follow the directions to complete the process to export a final APK file.
  ☐ `supplemental` – a directory containing any supplemental files required by your application that are not included in your Eclipse project. If you have no supplemental files you do not need to include this directory in this zip file.

☐ Finally, head to `https://www.cs76.net/submit`, click the **login** link at top-right, click the link to your TF's dropboxes at top-left, click this project's own folder, click **Upload File**, and upload your ZIP file as prompted; no need to give it a title. Be sure not to click on the wrong TF's dropbox or the wrong project's folder. You may re-submit in this same manner as many times as you'd like. Just take care to delete any prior submissions.

Be sure not to submit or re-submit after this project's deadline.