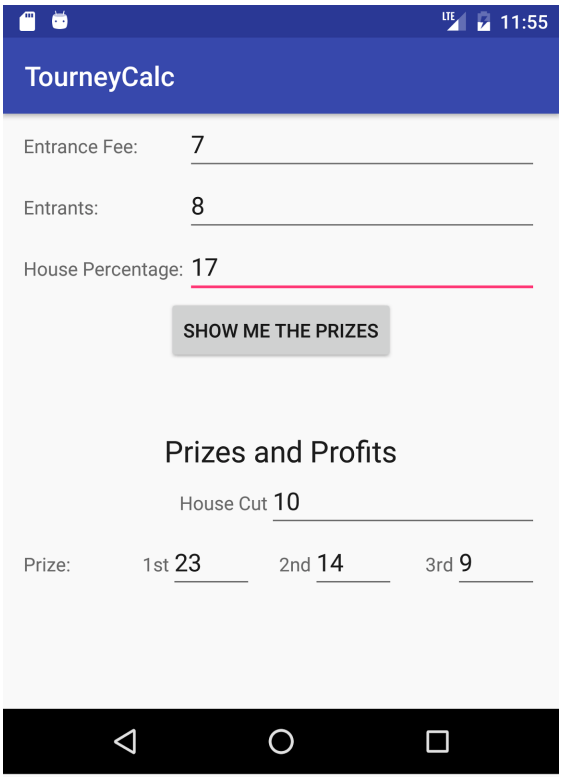
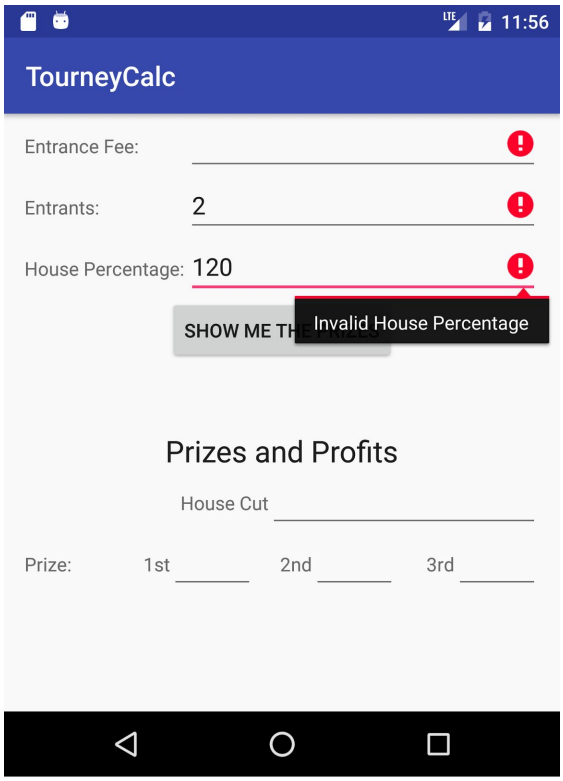
Assignment 4: Android Development

In this assignment you will build a simple Android app similar in spirit to the one shown at the end of the demo for the Android lesson. The app is a tournament prize pool calculator that:

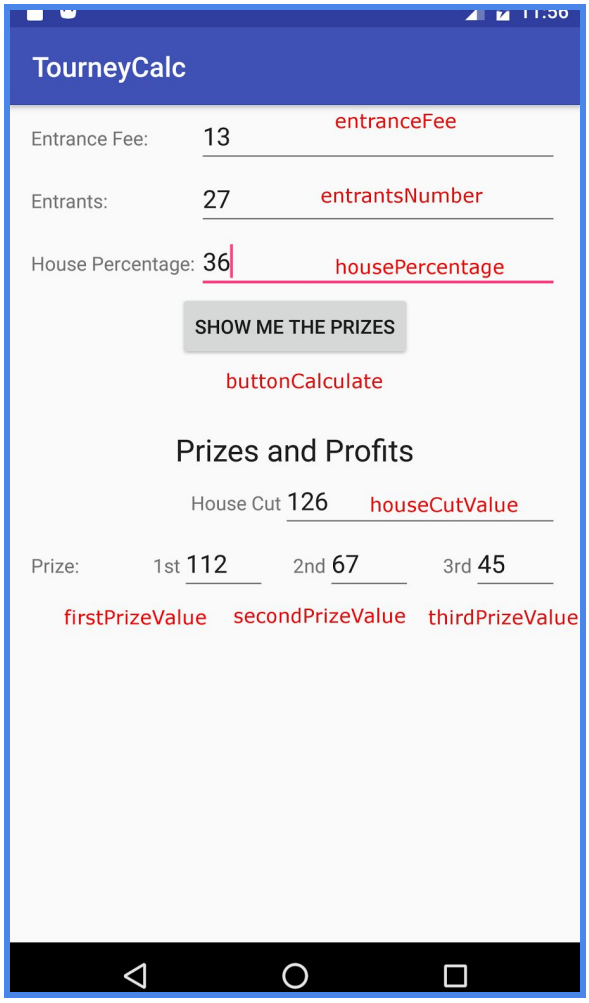
1. Takes as input
    a. the *entrance fee* (a positive integer)
    b. the *number of entrants* (an integer greater than 3)
    c. the *house percentage* (an integer percentage between 0 and 100)
2. Produces as output
    a. the *house cut*, computed as the *house percentage* of the total amount collected (i.e., *entrance fee * number of entrants*), rounded to the nearest integer
    b. the amounts assigned to *1st*, *2nd*, and *3rd prices*, computed as the 50%, 30% and 20% of whatever remains after the house cut is removed from the total amount collected, also rounded to the nearest integer (do not worry if prizes and house cut do not add up to the total–just assume the house will make it work)
3. Displays an error using a text view error message in case of **empty or invalid** input values (see Item 1 above for what constitutes correct input values). The error messages should be exactly "**Invalid Fee**", "**Invalid Number of Entrants**", and "**Invalid House Percentage**" for an empty or invalid *entrance fee*, *number of entrants*, and *house percentage*, respectively.

Here are two (slightly cropped) screenshots of the app that provide examples of inputs, outputs, and errors:





Please try to keep your UI similar to the one shown. Even if your UI is different, **please make sure to use the identifiers shown in the next figure for the key widgets in the UI**. This is very important, as we will use these identifiers to automatically test your app. The identifiers are also listed next to the figure for your (copy-and-paste) convenience.



Identifiers:

- `"entrantsNumber"`
- `"entranceFee"`
- `"housePercentage"`
- `"buttonCalculate"`
- `"firstPrizeValue"`
- `"secondPrizeValue"`
- `"thirdPrizeValue"`
- `"houseCutValue"`

For example, in the XML layout file for your app, the entry for the text field used to input the entrance fee should have the following ID: `android:id="@+id/entranceFee"`. For another example, the entry for the button used to trigger the computation of percentages and prizes should have the following ID: `android:id="@+id/buttonCalculate"`.

To help you make sure that you used the right identifiers, we are providing you with an extra source file that you can add to your app. The code in the file does nothing, but it prevents your app from compiling if your identifiers are incorrect or your activity name and package are incorrect. We provide instructions below on how to use this file.

To complete the assignment you must perform the following tasks:

1. In the root of **your individual GitHub repository**, create a directory called `Assignment4`. Hereafter, we will refer to this directory in your local repo as `<dir>`.
2. Using Android Studio, create an Android app project called “TourneyCalc” in `<dir>`. This should results in having a directory TourneyCalc under `<dir>`.
3. Create a single activity, called `TourneyCalcActivity`, that includes the functionality of the app and is part of package `edu.gatech.seclass.tourneycalc`. If you define additional classes, they should also be part of that same package.
4. Define the IDs for the key widgets in the app as described above.
5. Provide a `manual.md` file (in MD format) that describes how to use the app. Put the file in `<dir>`. Think of this file as a (very concise) user manual.
6. Perform the following sanity check on your project:
    - Download the extra source file we provide here.
    - Unpack the tar file you just downloaded in the root of your repo. This should add file `<dir>/TourneyCalc/app/src/main/java/edu/gatech/seclass/tourneycalc/SanityCheck.java` to your project.
    - Rebuild the project in Android Studio (it should happen automatically).
    - If the project does not compile, it should mean that there are issues with your activity name/package, with your identifiers, or both.
7. Commit and push your project from within Android Studio (or from the command line, if you know what you are doing). Doing it from Android Studio should help ensure that all the required files are committed. No matter how you commit and push your project, we strongly recommend that you clone your repo in a different directory and try to open the project in Android Studio, so as to ensure that you have included all the required files.
8. As usual, submit your solution by
    - Pushing your code to the remote GitHub repository.
    - Submitting the commit ID for your submission on T-Square. (You can get your commit ID by running "git log -1" and copying the hexadecimal ID it produces.)

Notes:

1. The minimum target for your Android app should be API Level 19.
2. Also in this case, you can perform multiple commits and work on multiple branches as you produce your solution. This is not only fine, but actually encouraged. Just make sure that your final solution is committed to the `master` branch.
3. You should complete the assignment using Android Studio 2.2 (or a later version).
4. Feel free to **take inspiration** from online resources when developing your app. However, be careful not to copy and paste entire pieces of functionality. The tool we use to identify cases of plagiarism is likely to have access to the same online resources that are available to you.