

University of Stirling
Computing Science and Mathematics
CSCU9YH

Project – Unit Conversion App

The Android Project is about a Unit Conversion application. There will be some lab sessions to help you on your way with some key elements of this application. However, you are expected to add to these in your own time.

In brief, the application is an app to convert between different units (distances, weights, speeds etc). The app will have two screens, one for selecting the two units to convert between, and the other to actually carry out the conversion. The two screens are to be implemented using Fragments within an Activity. **Before you continue with working on the tasks in this handout make sure you have completed the lab on the YH Project (on fragments).**

In previous labs you have created Activities, databases, a basic GUI, and in the project lab Fragments.

The project should include the following functionality:

- The first screen should allow the user to select the two units in a user friendly way, ideally using a spinner or even picker.
- The second screen is expected to look a bit like a calculator allowing the user to type in a value for the conversion. There should also be buttons to clear any previous input and to initiate the conversion.
- The second screen should also have an indication what units were selected on the first page.

At the most basic level, your application should:

- Support two fragments
- Be able accept user input for the given unit as well as the target unit (on the first fragment), as well as the value (on the second fragment).
- Convert the value into the target unit
- Offer a fixed set of at least four unit conversions (e.g. meter to yards, miles to yards, grams to ounces, kilograms to pounds).

Further functionality and refinement

The functionality listed above is what is expected of a basic app. However, for additional credit you can refine your application further. Rather than hardcoding the four conversions, you may consider using persistent storage for this data. You may also extend your application in such a way that the types of conversions your app can do is extendible (restrict yourself to simple conversions involving a single factor). Introducing categories of conversions (distance, weight, speed etc) will allow users to select the units more easily as the number of conversions offered

grows. A common feature many users expect is that the app also works correctly, when the screen is rotated and that the app also works nicely on larger screens, e.g. tablets. However, you may have your own ideas for refinement of the app.

How to begin

Start a new project and give it an appropriate name, such as UnitConv. Setup a new package within that project with a name such as uk.ac.stir.cs.yh._yourInitials_

Start with a minimal application and make sure that it runs. Then build up the application in small steps, making sure that it compiles and runs as expected after each step.

Submission Instructions

The deadline for handing this project is **10am, Monday, 25th November 2019**. Separately, you will be asked to demonstrate your solution to the given problem. Demonstrations will take place in the lab session on **Monday 25th November (possibly running into the Wednesday lab slot too)**. For the submission you should prepare a single document which includes a report discussing the problem, any assumptions you made, and **your solution**, as well as the code listings of your program (java classes, resources xml files). Make sure the source files are formatted appropriately and are readable. Your report should present your solution and discuss the structure of your app, key features, and UI design. The report should include appropriate diagrams of the design and screen shots of your application. The report also should provide details as to how complete your solution is, any additional functionality you have implemented, and if applicable, any special cases when your program is not working correctly. Make sure your report is laid out nicely and looks professional. Your report should be critical and reflective in nature. You may want to discuss alternative approaches to implementing certain functionality. Are there better ways to implement features, but you have not used these (e.g. due to time constraints)?

You should submit your document via Canvas. *You are expected to demonstrate your solution*, so please do test out the final version. Make sure that what you submit does work in some fashion. You can delete or comment out incomplete code before submission. After submission, you should **leave your files** in the folder untouched, until you are notified of your grade for this assignment. **At the demo**, you will also be asked for an electronic zipped copy of your assignment files.

It is important that your program code is thoroughly and intelligently commented. You will lose marks if the code is not sufficiently and usefully commented! Remember, do not state the obvious. Good comments explain the *use* of variables and methods/classes, and *why* certain values are assigned.

In short, your project consists of:

- a cover sheet giving your student number,
- a report discussing your solution, including your program code with comments,
- Separately, a demo of your solution + an electronic copy of your solution,

Plagiarism

Work which is submitted for assessment must be your own work. Students are encouraged to share ideas. However, each student must individually code their own assignments. Source code will be checked and tested to verify individual work. All students should note that the University has a formal policy on plagiarism which can be found at <http://www.stir.ac.uk/academicpolicy/handbook/assessment/#q-8>.

Plagiarism means presenting the work of others as though it were your own. The University takes a very serious view of plagiarism, and the penalties can be severe. Specific guidance in relation to Computing Science assignments may be found in the Computing Science Student Handbook.

We check submissions carefully for evidence of plagiarism, and pursue the cases found. Penalties range from a reduced grade, through a grade 5 for the module, to being required to withdraw from studies.

Assessment Criteria

In this assignment we shall be assessing your work with respect to various criteria, the most important of which are:

- Correctness of operation
- Appropriate use of programming constructs
- Intelligent code comments
- Clear and comprehensive report
- Consistency, legibility and tidiness of program layout

The marks for the project count for 90% of the module grade with the remaining 10% being allocated to the lab checkpoints. The split for the project is as follows: report (30%) and the code (70%). The Code component in turn is made up of basic functionality (40%), advanced features (40%) and code comments (20%). Advanced features include facilities to extend the conversions from the basic four, unit categories, more user friendly schemes to select the units: e.g. a spinner; enable phone re-orientation so that the app is viewable in portrait and landscape mode.

Late submission

If you cannot meet the assignment hand in deadline and have good cause, please see Dr Mario Kolberg to explain the situation and ask for an extension. Submissions will be accepted up to **seven days** after the hand in deadline (or expiry of any agreed extension), with the mark being lowered by three points per day. After seven days the work will be deemed a non-submission and will receive an X (no grade).

Backups: You are advised to make backup copies of your work regularly.