

# Inf2SE: Coursework

Nigel Goddard

School of Informatics  
University of Edinburgh

# Motivation

- ▶ there's no way to get comfortable with big systems other than practice
- ▶ students have different interests, backgrounds and levels of confidence, so it's best to let you pick individual projects
- ▶ by insisting that you must all use different projects, I also do away with plagiarism concerns and free you to collaborate: please do! Will explain shortly how that's to be handled.
- ▶ there's no good way to assess whether you have learned this stuff under exam conditions. Hence for-credit coursework.

Let's look at the document and sites...

# Practicalities

Can work on own machine if you have one. Backup is then YOUR responsibility.

If working on DICE, may need to use a USB stick or similar for large packages, so as not to run out of quota. (Save a copy of any file you modify into your home directory where it will be backed up, though!)

Probably best to start early, build it successfully, then sign up for it - please try not to bombard me with requests to change projects.

## What to do when build problems hit

- ▶ Read the error message carefully. It may just be that you're missing a library.
- ▶ RTFM, if there is one
- ▶ Google for the precise error message you get. This is a surprisingly good way to get a better idea of what causes it.
- ▶ Ask another student for help. Sometimes just explaining it to someone else will help you see the problem.
- ▶ Check what Java version you have and whether it's appropriate
- ▶ Try building from the command line instead of from Eclipse
- ▶ Attend an office hour and ask me for help (but don't expect miracles!)
- ▶ Ask on the Inf2SE bboard thread (read there too: you may learn from other people's questions, and if you give helpful suggestions you may get credit for it)
- ▶ Ask on the project's mailing list, or email the project owner

# Making a contribution

As the handout states, many different kinds of contribution are acceptable...

Part 1 Q3: when considering which would be best, make sure your argument is about **your particular project** not just projects in general.

Part 2: very open. Do what you want to learn about. Make sure what you do is software engineering though – e.g., don't spend hours on translating text. Write it up clearly but concisely.

Do try to feed your contribution back to the project if possible - many students did, last year.

# Submission

Submit carefully, in HTML, using the required file names, making sure image files are in the right place, etc. Try copying the directory you plan to submit to somewhere else in your file system, then viewing it in your browser, if you want to check you have it working.

You may exercise your HTML design skills, but are not required to. Your submission may be looked at on a netbook - don't use a fixed page width! (Resize your browser window with your submission in it, check it still looks OK.)

Quality of writing is much more important than quantity. No marks for grammar and spelling, but clarity is crucial.

# Feedback

You will get (by email) marks per question with brief comments.

If you had doubts about some particular aspect of your work and would like specific feedback on that, feel free to say so in your submission. (I may reply in the feedback email, or invite you to make an appointment afterwards if it's more complex).

If you don't understand the feedback or want to discuss the work further you're welcome to make an appointment to do so (but this is an invitation to discuss the work, not to dispute the mark).



## Finally, strategy

Given a choice between something you reckon you could get 8/10 for by applying what you already know, and something you reckon you might only get 6/10 for but will enjoy and learn things from, I hope it's a no-brainer which you should pick...

Even from a mercenary point of view, learning things now will pay you back in marks later, even if it costs the odd mark now.

And it probably wouldn't cost you marks now, anyway.

(In particular, I try hard not to give less than 4/10 to anyone who has kept up with the course and spent the recommended time in a reasonable way.)

It is supposed to be fun...