

CS3283 Assignment #3

20th March 2003

Due by 5pm on Wednesday, April 16, 2003
Delivered to Hugh

You may work in groups of (upto) four students. Do not bother asking for a group of five. The group members do not have to be from your tutorial group, and it is up to you to ensure that everyone contributes equally. Email Hugh with your group members before March 28th 2003.

This assignment is worth 40% of your assignment mark. It consists of the development or modification of a Java program, with some documentation.

Your task...

You have two possible tasks. You can do either one of them, but I should tell you now that the first is HARD, and I advise students *not* to take it up - I am giving it to you as a challenge:

Task #1: reimplement your Tcl/Tk assignment #2 as a Java program, using Swing.

Originally I intended this to be assignment #3, but on reflection, I think the problem is too hard, and I have decided to set a more conventional programming assignment, which does not have much tricky canvas manipulation. If you intend to take up this challenge, come and talk to me first.

Task #2: Your task is to implement and document a Java application which provides a user interface for a tool to assist in the management of moveable assets in a building, which I will attempt to describe below. This application is likely to display TEXT information only.

The interface is a special purpose file editor, intended to manage the process of editing and querying the moveable assets in a building. Moveable assets should at least include

- Computers
- Desks
- Chairs
- Filing cabinets

The user of your application will be using a main screen which shows a list of assets. This screen may just show a text listing of a file used to store the asset database, or it may be done in a spreadsheet like manner. The user of your application will be able to load new asset databases (which may of course just be files), to insert new databases, to create new, copy and delete database entries and to save back the resultant database. In addition you must have a help menu or button.

Each asset has editable information about that asset. The information should include:

- The date and time the asset was first entered into the system
- The room it was first allocated to
- The room it is currently in
- A unique identifier/brief description for the asset
- A scrollable text box with (say) 5 visible lines of text.

The main screen should allow you to display the information sorted (alphanumerically) by any one of these fields. For fields that are too long, you may truncate the field when you display it.

Field editing: When you select a particular line on the main display, a Java/Swing dialog box is to be created with editable fields for each of the information fields given above. After editing the fields, you can choose to save the data back, and this is reflected in the original screen.

Notes:

The following points form part of the functional specification of the user interface.

1. At any time you should be able to save and restore the state (i.e. the system is persistent)
2. The program should be safe on failure - that is - if some part of your program dies for some reason, you can re-run the program, and get back to where-you-were.
3. The minimum flow of operation of the interface is that you can
 - (a) add new assets,
 - (b) edit asset fields,
 - (c) save and load new databases,

Deliverables:

You are to present your assignment as a single (zipped) file containing the sourcecode and a README file outlining how to run your program, along with an electronic version of the documentation in PDF format. The documentation can be minimal, must also be presented on paper, and should contain:

- A title page containing your names and matriculation numbers.
- Table of contents...
- A one page introduction describing the application function in a brief non-technical style.
- A one page section describing the system, (such as how you store the database, file formats and so on).
- A one to three page section describing the interface design (what are the screens, what is the general flow of operation of each screen).

Note that this assignment *does* require you to *implement* the application, and it must be implemented in Java/Swing, and (preferably) runnable on the Suns.

Assessment:

The assessment is as follows:

Documentation	15%
Code style/quality	35%
Operation of the interface	50%

Try to achieve clarity in your writing and take care in the structuring of the document.

COOPERATING VERSUS CHEATING

You are allowed to discuss the problems with your friends, and to study any background material with them, but the assignment *should be your own group's work*. **Copying** and **cheating** will be grounds for failing the assignment.