

# COMP101 Lab2: First Assessed Coursework

## Giant Letters

Worth 12% of the final course mark.  
Assignment one (of seven assessed exercises).

Failure to submit this assignment, or the submission of work that is deemed not to be a reasonable attempt, will result in the failure of the COMP101 module.

Completion of the implementation and report as described below will obtain 90% of the marks. To obtain the final 10%, students should also complete the Extended Requirements. This may involve doing some additional reading beyond what has been presented in lectures and/or slightly more complex programming. Only attempt the extended requirements if you are confident with programming.

**Learning Outcomes.** This assessment (partially) addresses the learning outcome, “to be able to implement, compile, test and run Java programmes, comprising more than one class, to address a particular software problem”.

**Requirements.** Design and implement a Java application program that writes your Computer Science user name (not your password) vertically down the screen using giant letters made up of strings of \* characters and blank spaces (not the “tab” character) in much the same way that we wrote the words “JAVA” in the giant Java example program given during lectures. It does not matter whether the output uses upper or lower case letters.

**Extended Requirements.** The program should allow the use of a single character (which could be different for each of the giant letters you print) which will be used to compose a giant letter. In other words, if you wanted to compose a giant J using the letter J (instead of the \* character), your program should let you do that. Alternatively, if you wanted to use, say, the % character to make a giant J, that should also be possible. Note that you \*do not\* have to obtain user input.

(Hint: First think how you can allow for any character to be used for a single giant letter.)

**Submission Instructions.** Your submission should consist of a short report (in PDF format) and implementation file(s) (.java files).

- The report must be a PDF file and should consist of

Requirements: Summary of the above requirements statement.

Analysis and Design: A short (one paragraph) description of your analysis of the problem including decisions about the classes used, a Class Diagram outlining the class structure for your proposed solution, and pseudocode for the key methods.

Note that as this is a simple program, a single class should be sufficient.

Testing: A set of proposed test cases with expected results, and evidence of the results of testing (the simplest way of doing this is to cut and paste the result of running your test cases into your report or provide a screen shot).

- The implementation should consist of

Your Java source files, i.e. the relevant .java files, not the class (.class) files.

Upload your files to

<https://sam.csc.liv.ac.uk/COMP/Submissions.pl>

(you will need to log in using your Computer Science username and password).

**Submission Deadline.** Monday, 17 October, 5pm.

**Mark Scheme** Marks will be awarded for

- Analysis and Design 10%
- Programming Style 20%
- Functionality and Correctness 50%
- Testing 10%
- Extended Requirements 10%

Please see the module web page for the feedback form. Reports not submitted in PDF format will be penalised.

### Note

- Because submission is handled electronically, ANY FILE submitted past the deadline will be considered (at least) ONE DAY late. Work will be penalised for lateness in accordance with university policy.
- Please make sure your Java file(s) successfully compile and run on ANY departmental computer system, both Windows and Unix. For this reason, the use of an IDE like NetBeans or Eclipse is discouraged.
- Note this is an individual piece of work. Please note the University Guidelines on Academic Integrity (see [https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix\\_L\\_cop\\_assess.pdf](https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf)).