

Assignment Project Exam Help

<https://powcoder.com>

Add WeChat powcoder

Compilers and Computer Architecture (G5035)

Submission guidelines. Here are some recommendations for your submission. Your code should have the following properties.

- To assure anonymity, do not put your names anywhere in your submissions, but do include your candidate number somewhere, preferably as the name of the submitted file.
- Please make sure your submission isn't 'noisy'. In other words, when I run your code, it should not output debugging information. (Maybe look at all your `System.out.println (...)` statements: are they really needed?)
- Please make sure the code you submit can be compiled by invoking `javac *.java`. In particular, you'll have to include the interface definitions that you have been given. **Code that cannot be compiled will receive 0 points.**
- You will have to use the code templates given to you. Do not change the names and signatures of classes and methods given to you. Of course you may extend classes, create new classes, instantiate interfaces etc. But a program that is compiled against the Java fragments you've been given as part of the tasks should remain compilable with your modifications and additions.
- Note that classes you submit might be instantiated multiple times, and methods might get invoked more than once. They should behave appropriately when used multiple times.
- If your code encounters a problem, do throw one of the exceptions defined in the tasks with an appropriate error message. Do not throw other exceptions and don't exit the program (i.e. don't use `System.exit(...)` and the like).
- Please put the submissions for each task into separate directories (called `task1`, `task2`, and `task3`). Each directory should contain **all files** relevant for the task, in particular for compilation (where that makes sense). I will compile e.g. your submission for Task 2 by going to directory `task2` and then running `javac *.java`. **Code does not adhere to instructions will receive 0 points.** Please compile each task yourself before submission using `javac *.java` to ensure that your submission is compilable.
- If you don't want to submit a solution for a task, omit the directory for that task.
- Please put your whole submission into a single directory that has your candidate ID as name. This single directory will then have sub-directories `task1`, `task2` and `task3`.
- Please zip-archive or tar your submissions, containing everything you want to submit. Submit only that archive as one file. The name of your submitted file should contain your candidate number, e.g. `23472.zip` or `43923.tar`. Before you submit, please check that your archives can be extracted. Do **not** use `rar` to make your submission into one file: `rar` cannot be reliably un-rared on OSX and Linux.
- Your submissions should *not* define a class `Main` (or `main`). More generally, you should not submit any classes that define a `public static void main(...)` method. This is because my testing framework will supply a `Main` class with a `public static void main(...)` method to execute tests of your code.
- Eclipse and other IDEs use different directory layouts, so if you use Eclipse and the like, you will have to adapt the directory structure to the submission requirements.
- If you use an IDE such as Eclipse or BlueJ, please make double sure that your submission is in the right format. Some IDEs (e.g. BlueJ) may generate files with dependencies that I cannot resolve as I don't have the relevant libraries, but that are automatically resolved when using that IDE.
- I will run the tests using Java 8 (more precisely Java version "javac 1.8.0_31"). If you are planning to use Java features that are not available in Java 8, please contact me.
- Please do not use Java's package construct. I know that it is good Java style to use package, but using it makes evaluating your submissions more difficult for me.

Originality and plagiarism. Please don't forget that the work submitted must be your own. For example, it is completely unacceptable for two or more students to collaborate on a program up to the point where it is typed into the system. You may discuss the designs of your programs with each other in broad outline, but discussions at the level of program statements are clearly unacceptable. If you have not done so already, you would be well advised to consult the University of Sussex Handbook guidelines on the subjects of collusion and plagiarism: see [here](#).
