Parameters on Project Help

As discussed in the syllabus, the class will be held in a lab/studio format; discussion of weekly project progress at these times, or as arranged with individual lecturers. 3 to 4 projects are interlaced with the parallel course lectures. Each project spans 3-4 weeks and has intermediate, weekly deadlines.

While we lecturers provide a project problem and pointers to solve it, you will learn the techniques necessary on your own, by reading and studying. We will provide feedback and help you, but we will not tell you how to solve the project in detail. These are the detailed parameters we will employ when we help you.

We will be available for troubleshooting/debugging tips, but will not solve your problems for you. When coming for extra help, we expect you to supply:

- A clear statement of what you are trying to accomplish.
- A clear and concise description of a particular problem. “My code does not work” is neither clear nor concise.
- A list of things you have already tried before coming for help.
- Possibly (at our discretion) a minimal example of code that exhibits the issue you are having, including a pseudo-code description of the algorithm you are implementing.

We cannot read over a 200-line code and find the problem for you. You have to develop strategies to do that yourself, including “reducing code” to isolate an error.

If you receive advice and choose not to follow it, you are on your own.

Other requirements:

- Codes must be grammatically and syntactically correct, irrespective of the language used.
- Codes must be portable and standards-compliant. Non-standard or proprietary constructs are unacceptable.
- Codes must compile and run on any unix/linux machine using the gcc/g++/gfortran compiler. If you choose not to use gnu compiler collection, then you must compile according to industry standards. If not, your code will be returned to you ungraded.
- Whether or not you use gcc, you must code to ANSI and/or ISO standards.
- With C you can accomplish this in gcc by using one of the flags -ansi or -std=c99. Using -pedantic will give you both.
- With C++ and g++ you should use: -ansi, -std=c++98, or -std=c++03. Using -pedantic will give all.
- With Fortran and gfortran you should code to the 95 standard (-std=f95). Fortran 77 is obsolete and must be avoided.

It is your responsibility to make sure your code compiles when we test it. The above are the standards we apply.