

# All About Homework

The homework will consist of both programming exercises and written questions. This handout describes some general thoughts and techniques for doing homework, as well as what is required, how to submit it, how late homework is handled, and other administrative matters.

## Turning In Homework

All homework is due at 11:55PM on the due date, unless noted otherwise on the assignment. These will be graded and comments returned to you as quickly as possible; we'll try for three class periods, but can't guarantee it.

For written homework, you must turn in an ASCII, a PostScript, or a PDF version of your answers (you can use any text processor you like to generate these). Please do not submit Microsoft Word files; since the readers may grade these on UNIX-based and Linux systems, they will not be able to read those files. If you submit PostScript, please be sure the file will print on our department printers (use *ghostscript(1)* or *gs(1)* to check this; if it displays the file properly, the file should print correctly). If your file is a PostScript file, please choose a name that ends in “.ps”. If it is an ASCII file, please choose a name that ends in “.txt”. If your file is a PDF file, please choose a name that ends in “.pdf”.

For programs, turn in the source code, Makefile (you must include one) and any related information (such as man pages and README files). Bundle these into a *tar(1)* file, and submit it. If the file is an uncompressed *tar* file, choose a name that ends in “.tar”. If you compress the file using *gzip(1)*, choose a name that ends in “.tgz”. Be sure that we can recompile the program without errors by typing “make”. As we will grade them using systems in the CSIF, please check that your programs run on one of the Linux systems there. You are free to use any programming language that is available on the CSIF and that the ECS 153 graders can get to (but we prefer C, C++, or a UNIX scripting language such as *sh(1)*).

Please turn in your written exercises and programs electronically through MyUCDavis. Do not use the *handin* program! If you need to turn in something on paper (for example, a diagram that you can't draw using your text processing program), please hand it to the professor or TA before the assignment is due, and put a note in what you submit electronically that you have done this. (That way, the reader will know to look for something written, rather than mark you off for that problem.)

## Doing Written Exercises

When you are asked to analyze something, or explain something, please be complete, and show your work (including any commands you give, and their output, to show how you did the problem). Otherwise, even if you get the right answer, you will get **ZERO** (that's *0*, *zip*, *nada*, *rien*, *nothing*) points. Think your answer through and do a rough draft. Students (and professionals, actually) often overlook this, but it is vital. Write clearly and cogently. If the question asks for an opinion, state your opinion clearly, justify it, and don't ramble. Answers that start, “My opinion is yes ...” and conclude with “... on the other hand it could equally well be no” won't get much credit.

## Doing Programming Assignments

Please do not leave assignments for the last minute. The assignments are non-trivial and will require significant design time before you start programming and debugging. When we decide on the due dates, we assume you will spend significant amounts of time on design as well as coding and debugging. If you choose not to do this, you will have difficulty finishing the assignments on time.

Please take the time to design your program carefully. More programming problems arise from improper design than anything else, and the few hours you spend on design will be amply repaid by shorter coding and debugging phases. So please think the design and interfaces through, and—as always—try to find the simplest way to do the assignment (within the limits given in the assignment, of course)!

We do not mind being asked for help; indeed, we welcome it because it helps us know what students are finding difficult or confusing, and sometimes a few words about the problem in class will clarify the assignment immensely. We do mind being asked for help before you have tried to think the problem through. The classic objectionable question (this really happened) occurred on a homework assignment in which the class was given a buggy program. The assignment said the program did not work, and the homework was to debug it and make it work. That particular class period discussed how to deal with bugs, and gave tips and techniques on how to debug programs. Within 10 minutes of the end of the class during which the assignment was given out, the

instructor got this request for help: “The program doesn't run. What do I do now?”

So, before asking for help, please be sure that you have:

- spent a significant amount of time on the design of your solution;
- used a debugger if the problem is a programming bug;
- read all relevant handouts, and news articles (because your question may be answered there); and
- tried everything you could think of to solve the problem.

When you come to us, or send us a note, asking for help, please show us whatever you have done to solve the problem, because the first question we will ask you is “What have you tried?” This isn't because we think you're wasting our time. It's because understanding how you have tried to solve the problem will help us figure out exactly what your difficulty is and what we can do to help you. Remember, we will do everything we can to avoid solving the problem for you. When we give you help, our goal is to help you solve the problem yourself.

## What We Look For In Programming Exercises

When we grade your homework, we look for simplicity, clarity, elegance, and documentation. Here's a rough weighting of the various factors that go into the grade of each programming assignment:

Correctness.....	60%
Commenting, ease of reading.....	20%
Clean, readable output.....	10%
Documentation (README, man page, etc.).....	10%

If a program does not compile (or assemble), the most you can get is 30% of the value of the program (20% for commenting and ease of reading, and 10% for documentation). So check your programs before you submit them.

## Late Homework

You can turn in your homework up to one class period late (unless the assignment says otherwise). If you turn it in late, we will grade it normally, and then deduct 20% as a late penalty. Requests for exceptions will be handled on a case-by-case basis; please do feel free to ask!

## Grade Appeals

If you feel that there is an error in grading, please come see the professor or the TA and we'll look over it (and possibly talk with you about it). However, don't dally; any such request must be made within one week of when the grades were made available. After that, we won't change your grade.