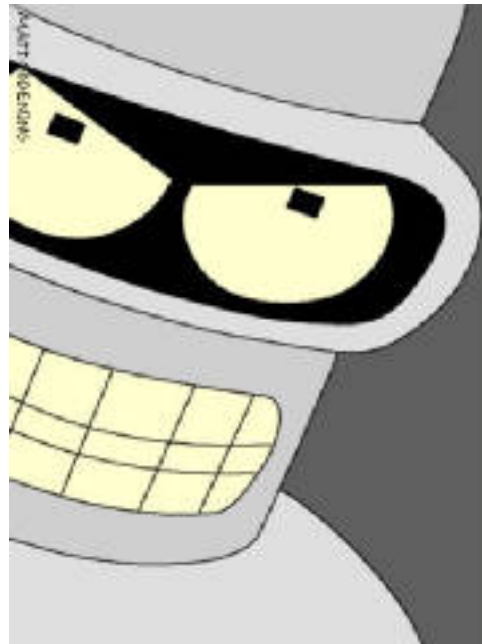


ECS 145, Scripting Languages and Their Applications

Norm Matloff

Spring 2009



**YOU WILL BE TESTED ON THIS
SYLLABUS!**

Contents

1	Crucial Importance of This Syllabus	4
2	Consultation	4
2.1	Office and Office Hours	4
2.2	E-Mail	4
3	Prerequisites	4
3.0.1	Unofficial Prerequisite	5
4	Course Content	5
5	Course Materials	6
5.1	Printed Lecture Notes	6
5.2	Each Student Must Have His/Her Own Course Materials	6
5.3	Bring the Course Materials with You to Class	6
6	Tests	6
6.1	Terminology	6
6.2	Exam Schedule	6
6.3	Quizzes	6
6.4	Open-Materials Policy	7
6.5	Work Only on the Test Sheet	7
6.6	Grading of Exams	7
6.7	How to Do Well on the Exams	7
6.7.1	Think, Not Memorize	7
6.7.2	Old Exams Yes, “Practice” Exams No	7
6.7.3	Homework-for-Learning Yes, Homework-as-Practice-for-Exams No	8
6.7.4	Concentrate During Class	8
6.8	Ethical Behavior Is Central	8
7	Lecture Format	9
7.1	Bring the Materials to Lecture	9
8	The Discussion Section	9
9	The Class Newsgroup and Web Site	9

10 Grading	10
10.1 Weighting	10
10.2 + and - Grades	10
10.3 Exams Grade	10
10.4 Quizzes Grade	11
10.5 Homework Grade	11
10.5.1 You Must Turn in All Assignments to Pass the Course	11
10.5.2 You Should Get an A on the Homework	11
10.5.3 Back Up Your Files Frequently!	12
10.6 Accuracy of Records	12
11 Machines	12
11.1 Linux PCs in CSIF	12
11.2 Your Own PC at Home	12
12 Homework	12
12.1 Terminology	12
12.2 Announcements	13
12.3 Partners	13
12.4 Programming Style	13
12.5 Submitting the Homework	13
12.6 Grading	14
12.7 If You Get Stuck	14
12.7.1 We Are Happy to Help If You Have Reached an Impasse	14
12.7.2 Use a Debugging Tool!	15
12.7.3 When in Doubt, Check It Out!	15
12.7.4 Help by E-Mail	16
12.7.5 Start Early!	16
12.7.6 Homework Ethics	16
12.8 Late Work	16
12.9 Reading Homework	17
13 I Do Care!	17

Things to Take Careful Note of NOW:

- You must notify the TA of the membership of your homework group by Friday, April 3.
- Homework I, Part A will be due on Tuesday, April 7.
- This course will require some use of Linux.
- The course focuses on principles, not syntax.
- You will be required to learn some material, e.g. advanced matrix operations and other mathematics, on your own.
- There will be many Tests—weekly Quizzes, and three Exams.
- You will be tested on the contents of this syllabus.

1 Crucial Importance of This Syllabus

Any Test may contain questions on the contents of this syllabus.

Note that this syllabus, and lots of other important information, is available at

<http://heather.cs.ucdavis.edu/~matloff/145.html>

2 Consultation

2.1 Office and Office Hours

My office is in 3053 Kemper, Ext. 2-1953. My office hours will be Thursday, 2-3. **I enjoy my office hours very much, and look forward to interacting with you during them.**

I am also available at other times if you have short questions.

Our TA will announce office hours soon.

2.2 E-Mail

You are welcome and encouraged to send me your questions via e-mail. I read my mail every day, including evenings and weekends.

3 Prerequisites

The prerequisite is programming ability at the level of ECS 60. It is also assumed that you know calculus and linear algebra.

Though Python is a cross-platform language, some homework assignments and Test questions will involve Linux. You should be reasonably good at programming, and **have a working knowledge** of basic Linux concepts such as

files and directories, e.g. commands such as **ls**, **cd**, **pwd**, **chmod**; processes, e.g. use of **ctrl-z**, **&** and the **bg/fg** and **kill** commands; shell conveniences, e.g. file name wild cards, the **history** command, etc.

Again, it will be assumed that you have a **working knowledge** of the Linux tools mentioned above. If you are lacking some of them, please see my UNIX tutorial Web page, at <http://heather.cs.ucdavis.edu/~matloff/unix.html> If you do lack this background, be sure to take a look at these tutorials **NOW**, before it is too late to remedy any deficiencies. Remember, these are tools which you will be using every day in our course, and **they may arise in Test questions**.

It is also worth mentioning that Intel has complained that “UCD grads don’t know UNIX well enough.”

3.0.1 Unofficial Prerequisite

An important unofficial prerequisite is the willingness to learn on your own.

You will be asked quite a bit to learn on your own. In other words, not every aspect of, say, Python will be covered in class or in the handouts. You will be expected to be resourceful, looking on the Web or in books for some of the language constructs you’ll need. See my Python Web page, <http://heather.cs.ucdavis.edu/~matloff/python.html>, for some useful links.

4 Course Content

The term *scripting language* has never been formally defined, but here are the typical characteristics:

- Used often for system administration and “rapid prototyping.”
- Little or no typing of variables, e.g. no distinction between integer, floating-point or string variables.
- Lots of high-level operations intrinsic to the language, e.g. stack push/pop.
- Interpreted, rather than being compiled to the instruction set of the host machine.

We will focus on the following languages:

- **Python:** We will spend most of our time on this language, for its clarity and elegance, and due to its popularity. (E.g. Python is big at Google.)
- **R:** Some of you may know this as a statistical language, but it is more general than that, as a data manipulation and graphical analysis tool. (We will not do much with its statistical side.) It is definitely a scripting language, and includes some Python-like features (in some cases out-Pythoning Python).

We may spend some time on Perl as well, though I don’t like the language.

PLEASE NOTE THAT THE FOCUS IS ON PRINCIPLES, NOT SYNTAX. We are interested in the *why* and **how** here—why a particular language was designed the way it is, what the goals were, how it works inside, etc. The students will learn syntax largely on their own.

5 Course Materials

5.1 Printed Lecture Notes

There is no bound textbook. Instead, the text consists of my printed lecture notes (PLN), which will be handed out at various times in the quarter. They will be available (gradually) in class distribution and online at <http://heather.cs.ucdavis.edu/~matloff/145/CourseMaterials.html>.

5.2 Each Student Must Have His/Her Own Course Materials

Note that Tests are open-book, and there is no sharing of books or other materials (including calculators) during Tests. Thus every student must have his/her own copy of the PLN.

5.3 Bring the Course Materials with You to Class

Bring the PLN to lectures, discussion sections and Tests.

6 Tests

6.1 Terminology

- the term *Exams* refers to examinations given in lecture; see Section 6.2
- the term *Quizzes* refers to examinations given in discussion section; see Section 6.3
- the term *Tests* refers to the Exams and Quizzes

6.2 Exam Schedule

There will be three exams, with Exams I and II approximately at the 4-week and 7-week points, and Exam III on the last day of class, June 4.

Note; There is no final exam. Exam III is not a final exam; it only covers the material presented after Exams I and II, i.e. is not cumulative in the ordinary sense of the word. However, keep in mind that almost any computer science class is cumulative in the sense that earlier material may provide foundation for questions whose primary focus is on new materials.

There are no early or late exams. Provisions for oral makeup exams will be made if you have a medical excuse.

6.3 Quizzes

A Quiz will be given almost every week in discussion section. Quizzes will not be announced; assume that there will be one each week.

Quizzes will be short (10-20 minutes) and simple. The latter means that **Quiz problems will be very straightforward and much easier than Exam questions.** Their purpose is to ensure that you are keeping up with the material.

6.4 Open-Materials Policy

TESTS ARE TAKEN ON AN OPEN-MATERIALS BASIS. Bring the PLN with you to each exam. You are also welcome to bring whatever other materials you wish, e.g. your Linux books, dictionaries,¹ whatever you want. Whatever you bring, **make absolutely sure that you remember to bring all your course materials, as many of the exam questions will refer to specific pages in them.**

Please note that in program code on the exams you are only allowed to use Python/R/Perl/etc. constructs presented in our course materials, i.e. the PLN, handouts, announcements I make in the newsgroup, etc.

6.5 Work Only on the Test Sheet

Please work **ONLY** on the Test sheet, front and back. Do **NOT** turn in any supplementary sheets of paper. **A DEDUCTION OF 5 POINTS WILL BE MADE FOR NOT FOLLOWING THIS POLICY.** Please, it's important.

6.6 Grading of Exams

An Exam is supposed to be a measuring instrument, assessing the student's insight into the material. So for example, if a student's solution has random symbols and code which seem to match about 50% of the correct solution, it does not necessarily mean that the student will get 50% of full points for a certainly problem. Instead, the student will be assigned a score commensurate with my assessment of the student's insight. The assigned score could be either more or less than 50%.

An example of this point is that, although generally in the case of programming problems, code that works correctly will receive full credit, in some cases I might give less than full credit. It could be, for instance, that the code works correctly only "by accident," with two errors compensating each other.

6.7 How to Do Well on the Exams

6.7.1 Think, Not Memorize

The Exam questions probe more deeply than the Quizzes, testing insight, not memorization. They assume *active reading* and *thinking* about the material.

If for example you encounter a statement in the reading which puzzles you, don't just skip it—ask me or the TA about it. An Exam question may later ask you to explain it.

6.7.2 Old Exams Yes, "Practice" Exams No

Please note carefully that I do not give practice exams.

This is because I believe that learning does not consist of memorizing, and because this course (in the way I teach it) is not like a "calculus-style" course:

¹For example, do you know what the words *relinquish* and *discrepancy* mean? These words have shown up in problem statements in exams. I will be happy to explain any such words if you like during the exams, but you may feel more comfortable with a dictionary.

Calculus courses can have practice exams because the real exams have predictable, technique-oriented content. In a calculus class, you might have a lecture on integration by parts, then have a homework assignment of integration-by-parts problems, and then on the exam there would be a couple of such problems.

This would be impossible in this course—at least the way I teach it—because every Exam problem is unique, and the problems test understanding of concepts, rather than techniques. Thus, it would be impossible to have practice exams.

I do make my old Exams available on the Web, but they are NOT practice exams. Instead, I make them available so you can get an idea of the general “flavor” of my exam philosophy. Also, sometimes during a lecture I do mention a “focus area” that could be featured in an upcoming exam. For instance, I may say, “I really want you to know this particular example especially well,” and then the Exam may contain a question (or even more than one question) on this example.

6.7.3 Homework-for-Learning Yes, Homework-as-Practice-for-Exams No

Similar to the point that I do not give practice exams, it is important to understand that the homework is not considered practice for Exams either. The goal of the homework is to give you a chance to think more deeply about the material, and to give you a chance to nonpassively implement some of the course concepts. Sometimes an Exam problem will indeed be similar to some homework problem, but most Exam problems do not look like any homework problem.

Again, the best way to prepare for Exams is to constantly sift the material through your mind in a nonpassive manner, thinking about questions involving How?, Why/Why Not? What is the goal? and so on. The homework will really help you prepare for Exams in this sense, because it will make you think thoroughly about the course material, but you should not expect problems to be explicitly similar to the homework.

6.7.4 Concentrate During Class

MAKE SURE TO LISTEN CAREFULLY IN CLASS.

Weird statement! What does it mean?

Many Exam problems will come from the answers I give to questions students ask during lecture, or will be related to off-hand remarks I make during lecture when I suddenly get an inspiration.

Therefore, listen carefully to questions students ask in class, and to the answers I give. Make a note of seemingly-casual comments I make in lecture, and think about them carefully at home. These things may well pop up in Exams. Such questions will still be answerable from the course materials if you miss class, but there is a definite advantage to paying close attention in class.

6.8 Ethical Behavior Is Central

An embarrassing subject which nevertheless must be mentioned is cheating on Tests and homework. **If a student is found to be cheating, it will be treated as a VERY SERIOUS matter, not a harmless prank; it will harm his/her standing at the university, and also possibly make it difficult for him/her to get a job when seeking employment after graduation.**

Please work as follows during Tests:

- Try to sit in alternate seats if possible.

- **ABSOLUTELY NO TALKING** to classmates at any time during the Test.
- Keep your Test paper covered when you are not actually writing, so that it will not be so exposed to view.
- **IT WOULD BE GREATLY APPRECIATED IF YOU USE THE RESTROOM BEFORE THE EXAM STARTS, RATHER THAN DURING THE EXAM.**

During a Test, if your paper is exposed, I or the TA will come to you and mention this to you. Thanks very much for your help.

As to homework, you are welcome to discuss the programming assignments with your classmates, and to search the Web if you wish. But copying of code—ranging from a full program to a single loop—is unethical and will be prosecuted.

7 Lecture Format

7.1 Bring the Materials to Lecture

Make sure you bring the PLN to every lecture. They will play very active roles in the lectures.

I typically will ask you to open to a certain page, and then will call your attention to various sentences, examples and pictures on that page. The lectures will consist mainly of discussion and amplification of the examples in the PLN, with additional examples and comments. I will seldom write on the blackboard.

A MAJOR GOAL OF THE USE OF PRINTED LECTURE NOTES IS TO GIVE STUDENTS A CHANCE TO ASK QUESTIONS AND TO ENGAGE IN CLASS DISCUSSION. You are very much encouraged to bring up any questions you might have. Please don't be shy about asking questions; there is no such thing as a "dumb" question. Similarly, listen carefully to the questions the other students ask; this can be a very valuable source of insight, to which you should pay special attention.

8 The Discussion Section

THE DISCUSSION SECTION IS REQUIRED. It will be used for the purpose of giving Quizzes, and for presenting material not covered in lecture. (That material will appear on Tests.)

Please note that the purpose of the discussion section is NOT use as a forum for asking questions on how to do the homework. As stated in Section 12.7, the TA and I are happy to help you with the homework, but this should be done on an individual basis, not as a class. Please ask your questions either in office hours, by e-mail or individually after lecture or discussion.

9 The Class Newsgroup and Web Site

The name for the class newsgroups will be **ucd.class.ecs145** and **ucd.class.ecs145.d**.

THE NEWSGROUP ucd.class.ecs145 WILL PLAY A MAJOR ROLE IN OUR COURSE. IT IS REQUIRED THAT YOU READ THE NEWSGROUP EVERY DAY. Not only will announcements of homework assignments be posted there, but also there may be later corrections, hints, etc. And of course, dates of Exams and other related information will be announced on the newsgroup too.

You can read newsgroups using special-purpose programs such as **tin**, **slrn** or **pan**. Personally I use **slrn** and sometimes **pan**; you may prefer the latter because it has a nice GUI. Just type the program name, e.g. type **tin** when logged onto one of the CSIF machines. Or you can access newsgroup through general-purpose programs such as Web browsers.

Note that unless you know how to authenticate from non-campus locations (see <http://email.ucdavis.edu/newsgroups/reading.html>) the newsgroup can be accessed only from a UCD Internet address. You will have this if you are using a campus computer or have connected to the Internet via the campus modem bank. If you are not in this setting, you can still read the newsgroup by using **ssh** to log on to one of the CSIF machines. and then typing **tin** or **slrn** at the Linux command line.

Please note that “.d” newsgroups are for discussion among students. This was the original intention when they were established some years ago. It seems that instructors of ECS 30 and 40 have abandoned that concept, but I feel it is important for the students to have a place of their own, without interference from the instructor.

Thus I usually do not read the .d groups. If there is a message you want me to see, please send me e-mail, instead of posting a message which you hope I will see on the .d group.

In short:

- The .d newsgroup is for posting by students.
- The main newsgroup is for posting by me (and the TA).
- If you have a question for me, send it to me via e-mail. I will reply to you, and if I believe the topic is of general interest, I will post it to the main newsgroup.

10 Grading

10.1 Weighting

The formula used is

$$\text{course grade} = 0.6 \times \text{Exams grade} + 0.2 \times \text{Homework grade} + 0.2 \times \text{Quizzes grade}$$

where the Exams, Homework and Quizzes grades are each on the 4-point scale (4 for an A, 3 for a B, etc.).

10.2 + and - Grades

The threshold for a grade of n is (n-1).85; the threshold for an ‘n-’ grade is (n-1).6; the threshold for a ‘+’ grade is n.2. So, for example, if your weighted average from the above formula is between 2.6 and 2.84, your course grade is B-; if the average is between 2.85 and 3.19, your course grade is B; if the average is between 3.2 and 3.59, your grade is B+.

10.3 Exams Grade

Each exam grade will be used in letter grade form only (the numerical grades are recorded but not used).

IMPORTANT NOTE: If your first exam letter grade is lower than those of the other two Exams, and if your Homework and Quizzes grades are at least B, that first exam grade will be thrown out completely. In that case, your Exams grade will be the average of your second and third Exams. **This, coupled with the fact that most of you will get A's as your Homework and Quizzes grades, typically makes an important difference in the course grade for many students.**

By the way, though, please note that **this policy of course does not imply that you should not try very hard on the first exam**, for many reasons:

- It might turn out that you do very well on the first exam, better than one of the other two, so you definitely would want to have the benefit of that first high grade.
- Even though Exam I is the only exam which I **guarantee** that I will throw out if the other two have higher letter grades, I also may throw out Exam II or III. If someone does quite well on Exams I and III, for instance, but poorly on Exam II, I may either greatly reduce the weight of Exam II for that person or even throw out Exam II entirely.
- When a student asks me for a recommendation to an employer, or asks me to write a letter for graduate school, I take ALL Exams into account.

Exam grading is noncompetitive (there is no “curve”), so it is possible for everyone to do well.

10.4 Quizzes Grade

You will get a letter grade on each Quiz.

In recognition of the fact that on (rare) occasions you will not be able to attend discussion section, your two lowest (letter) Quiz grades will be thrown out. Your other Quiz letter grades will be averaged to produce your overall Quizzes grade, using the scheme in Section 10.2.

It is assumed/intended that, similar to the homework (see below), **most of you will get an A grade overall for your Quizzes.**

10.5 Homework Grade

10.5.1 You Must Turn in All Assignments to Pass the Course

<p>Note that all homework assignments must be completed in order to get a passing grade in the course. If you are missing any assignment, you will be given either an F or an I grade.</p>

In order to be considered “completed” the basic work must be done, though there may be errors. In the case of a programming assignment, for example, if the program has sections obviously missing, or if the program does not even compile, it is not considered complete.

10.5.2 You Should Get an A on the Homework

I aim for the vast majority of the class to get an A on the homework. Lots of help is available, so this aim should be achievable. (And it is; I use this policy in every course I teach, and it always works out that most people get A homework grades.)

The most important part of getting a good homework grade is to **start an assignment as soon as it is assigned**. Don't wait until a few days before the due date to start. **And make sure you have read the PLN thoroughly before starting.**

10.5.3 Back Up Your Files Frequently!

Accidental deletion or destruction of your homework files will generally not be grounds for extending the due date.

10.6 Accuracy of Records

At the end of the quarter, it will be your responsibility to make sure that the TA's records are correct as to your homework scores.

11 Machines

Our languages are available free for Linux, Windows and Mac platforms. In theory, you should be able to do most of your homework on any of these platforms. However, **some assignments will be Linux-specific.**

11.1 Linux PCs in CSIF

Our base machines will be the Linux PCs in the CS instructional labs (CSIF). If you pre-enrolled in this course, or if you took a CS course last quarter, you should already have an account. Your account name will be your surname, and if that is not unique, it will be your surname followed by your first initial. Your initial password will be the last eight digits of your student number; change it immediately, using `yppasswd`. Note that all the instructional machines share a common file system, so you can access the files from any machine.

11.2 Your Own PC at Home

I strongly encourage you to install Linux on your PC, if you have not already done so.

Moreover, having a Linux system at home will help you with your homework in many future CS courses, and will help you do well in job interviews when you graduate.

I have a Linux installation tutorial (though in need of some updating) at <http://heather.cs.ucdavis.edu/~matloff/linux.html>

12 Homework

We will have approximately three or four homework assignments through the quarter.

12.1 Terminology

- the *due date* means 11:59 p.m. of the stated date

- the *TA* means the TA in our case
- a *homework group* refers to a group of students working together on the homework (see Section 12.3)

12.2 Announcements

Homework assignments will be announced in the class newsgroup. The homework files themselves will be on the course Web site, so look there when an assignment is announced.

Occasionally there will be news about a current homework assignment, such as clarifications, hints and so on. These will be announced in the class newsgroup.

12.3 Partners

You are required do your homework together with two partners, for a group size of three; groups of size four will be allowed with special permission from me. Your group submits the homework together.

You are REQUIRED to submit your group membership list to the TA by Friday April 3 (11:59 pm. Anyone who has not informed the TA of his/her group membership by then will be assigned to a group by the TA. Subsequent changes in group membership will be approved only under unusual circumstances.

12.4 Programming Style

Good programming “style”—good indenting, meaningful variable names, helpful comments—is extremely important. You will not be graded on style, but it is crucial for communications purposes. For example, suppose you are having trouble getting your program to work and you e-mail it to me for help. **If I can’t read it, I won’t be able to help you.**

Note that this means that you write your program in good style **from the beginning**. Do NOT insert indentation, comments, etc. only when you are finished with the program and ready to turn it in.

The same points apply to top-down (i.e. modular) programming. Do this not because it is required for grading (it isn’t), but instead because it will make it easier for YOU, i.e. make it easier for you to write and debug your program, and easier for me to read if you ask me for help.

By the way, speaking of programming style: You may have been taught that global variables are “evil,” to be avoided like The Plague. That is far too extreme an attitude, and the reasoning used doesn’t make much sense. Moreover, in many cases one must use globals. See a discussion on this at <http://heather.cs.ucdavis.edu/~matloff/globals.html>.

12.5 Submitting the Homework

Since we use interactive grading (see below), you do not “turn in” your homework in the traditional way. However, you do have to send the TA an e-mail message of the following nature, to certify that you completed the work before the deadline.

- E-Mail message title:
Title your e-mail message “HwknX,” where n is the homework number in Roman numerals and X is a string consisting of the first four letters of each surname in your group, in alphabetical order(of

course, if a surname is less than four letters long, just use the whole surname). For instance, if the surnames of your group members are Crocker, Hopkins, Huntington and Stanford, then the title of your e-mail message for submitting Homework II would be HwkIICrocHopkHuntStan.

- Package your files:

When you have finished all of a given homework assignment, first use the Linux **tar** command to collect all your files for that assignment into one **.tar** file, say named **Hwk2.tar**.

- Timestamp your package:

Then run the Linux **sum** command on the **.tar** file to produce a checksum, which is kind of a code number which is (for all practical purposes) unique to that file. E-mail that checksum (two numbers, actually) and your **.tar** file name (but not the file itself), to the TA. The date and time on your e-mail will verify that you sent the mail before the deadline. Then later during the grading, the TA will run **sum** again, and the fact that the checksum has not changed will show that you have not changed the file subsequent to sending the e-mail.

12.6 Grading

We will use interactive grading. The TA will announce homework grading times, and each homework group will sign up for a time slot. **In the case of group size of two, both members of the group must be present during the grading time. If a member misses that time, the other group member will be graded, but the one who misses the time must reschedule a time in order to receive credit for the assignment.**

During a group's time slot, the group will log in to their account to access their files. They will then explain their homework solution, and in the case of a program demonstrate that it works. In the case of analytical homework, the group will explain the reasoning behind the work. The TA will ask each member of the group questions to confirm that he/she understood what they did. For example, such questions might ask why you wrote your program in a certain way, what the roles of the variables are, what numerical values the variables may have at times specified by the TA, why you did not choose certain alternative designs, and so on.

IMPORTANT NOTE: Each partner must be ready to answer questions about ALL the submitted work. This is because the two partners are supposed to work together on all the homework problems, which is very important. As a means of insuring this, **the TA will assign separate grades for each group member. Normally these grades will be identical, but if there is a substantial disparity in the levels of understanding the different group members have regarding the assignment, the TA will assign different scores to each member.** As mentioned earlier, if this occurs repeatedly, the TA will not allow the partners to submit work jointly anymore.

For each homework program, the TA will assign Extra Credit to the best programs submitted (up to a total of three programs, his call). The criterion to be used is clarity and elegance of the code; of course, the program must also work correctly. Addition of extra features will NOT be considered as a criterion.

12.7 If You Get Stuck

12.7.1 We Are Happy to Help If You Have Reached an Impasse

Please note that, **unlike ECS 30/40, homework assignments here will NOT lay out a detailed recipe, with tons of hints, telling you how to do the programming problems.**

The work is expected to be challenging, and is designed to be thought-provoking. This thought-provoking nature of the assignments is the vehicle by which you get to really understand the concepts. You are not necessarily expected to see right away how to do an assignment. Instead, you are expected to spend a considerable amount of time pondering the assignment, gradually seeing more and more, until you finally see how to do the whole thing. It is that thought process which will develop insight into the course material.

The TA and I will be quite happy to help you, including giving you hints individually—but only if, after giving a matter considerable thought, you still don't see what to do. Once you have reached the point where you cannot go any further, we very much encourage you to seek help from us. We want you to do well on the homework!

12.7.2 Use a Debugging Tool!

You will help yourself greatly—both in this course and in all courses in which you do programming—by making good use of a debugging tool, such as DDD or WinPDB. The debugging tools are described in my Python introduction, <http://heather.cs.ucdavis.edu/~matloff/Python/PythonIntro.pdf>.

Make sure you understand the content of my debugging slide show, at <http://heather.cs.ucdavis.edu/~matloff/debug.html>. Note carefully that what is important here is not merely **how** to use DDD, but **what you should use it for**. The latter consists of two categories of uses:

- (a) use in determining where in your code a seg fault is occurring
- (b) use in confirmation

Category (b) here refers to confirming that each part of your code is working as you expect it to: You step through the code line-by-line, confirming that the variables have the values you expect them to, confirming that the condition in an **if** statement hold and don't hold, according to your expectations, etc. Eventually you will find a place where your expectation is not confirmed—and that will be a big hint as to the location and/or nature of your bug.

Keep in mind that in situation (b), the debugging tool will not “automatically” find the location of your bug. You yourself must do that, through the confirmation process, but the debugging tool makes this far more convenient and thus saves you a lot of time.

Do yourself a big favor: Get good at using DDD, both in this class and in others, and you will reap big benefits.

When you seek help from me or the TA, the first thing we will ask you is to show us your debugger output. For Python, that will mean PDB; DDD and WinPDB are nicer, and you should use them, but you'll need a text-based debugging session to send us, so re-do the debugging steps in PDB and send us that output. Make SURE you use the **script** command to record your PDB session; see Section 12.7.4.

12.7.3 When in Doubt, Check It Out!

A lot of questions are of such a nature that you can answer them on your own, just by trying it out on the computer. For example, suppose you have a question which you would like to ask which is something like “Does Python allow such and such?” You are welcome to ask me or the TA, but you will get your answer a lot faster if you simply try it out! Put that instruction in a small test program, and see if the interpreter

accepts it or not. And if you are not sure whether the action does what you think it does, run the program and see!

Python's interactive mode is especially useful for quick tests like this.

12.7.4 Help by E-Mail

Again, you are welcomed and **strongly encouraged** to send me e-mail, at the address `matloff@cs`, for help on homework assignments. But you can help me if you keep the following in mind:

- **please do NOT send me e-mail in HTML format, or with lines longer than 80 characters.** If you do, I will of course still answer it, but my answer will not be as clear or as detailed. This is because when replying to e-mail, I like to quote parts of the original message, making the reply clearer. I cannot do this easily if your mail is in HTML or has long lines.
- Make sure you have clear indenting and commenting. (Again, for your own benefit, do NOT add these at the last minute; use them from the very beginning as you write your program.)
- Make sure you've made a good effort to track down your bug using DDD.
- If you send me e-mail asking what to do about some error message, please use the Linux **script** command to make a complete record of the error and the environment in which it occurred. Run both the command which gave you the error message, run PDB to try to get clues, and then run **ls -l** and **printenv**.

Take the output of **script**, together with your program source files and Makefile—and especially your PDB output—and package them into a **.tar** file, and e-mail me that **.tar** file. **Please make SURE to send me all of this in the form of a SINGLE .tar file.**

If you do not know how to run the **script** command, see <http://heather.cs.ucdavis.edu/~matloff/UnixAndC/Unix/UnixBareMn.html>

12.7.5 Start Early!

Please make sure to start the homework soon after it is assigned. Don't wait until the last couple of days before the due date, because **when the due date approaches, I tend to be inundated with e-mail messages asking for help on the homework, and thus my answers at that time will necessarily be short and without details, since I won't have time to say more.**

12.7.6 Homework Ethics

As discussed in Section 6.8, it is OK if you discuss the homework with students not in your group, either in person or in the `.d` newsgroup. **However, sharing specific code is not allowed.**

12.8 Late Work

An assignment is late if it is submitted to the TA after the due date.

If—this should never happen, but just if—you are late, you will be assessed a 5% penalty the first late day, and 10%-per-day penalty after that in your grade for that assignment. (Since e-mail is available every day, each of the seven days of the week counts as one day.) The maximum total penalty is 50%.

Each project group will be allowed a total of 2 late days over the quarter, time which is not penalized. You can use this as being late 2 days with no penalty on one assignment, or as being late 1 day with no penalty in each of two assignments.

In order to use this option, you must submit your work during the time you are using as your grace period. If for example you are 3 days late on an assignment, you will not be allowed to use your grace period time on that assignment; it will simply be counted as 3 days late.

Don't squander your grace period days in the first assignment. The subsequent ones will almost certainly be more difficult, so save your grace time for then.

The TA will keep the appropriate records as to how many days of grace period you have used..

12.9 Reading Homework

The reading—in the PLN—is extremely important, and should be considered “homework.” As mentioned earlier, approximately 30-50% of the exam questions will come from the reading.

After each lecture, do the corresponding reading. **Do the reading PROMPTLY, that same day, rather than getting behind.** This is the only way that you can get any insight from the reading; doing the reading the night before an exam will not work. **If you still have not done the reading as of the night before the exam, you actually will be better off not doing it, and getting more sleep instead, as cramming overnight won't be beneficial.**

13 I Do Care!

I wish to emphasize that I care very much that you succeed in this course, and I look forward to getting to know all of you.