IS 230 – Computer Programming I
Course Syllabus and Calendar – Fall 1999

Professor Don Colton

Brigham Young University—Hawaii Campus

Abstract

- **Course Number:** IS 230
- **Title:** Computer Programming I
- **Course Description:** An introduction to computer programming. Emphasis on fundamentals of structured programming design, development, testing, and implementation. Basic control structures of sequence, selection, and iteration. Sequential file processing.
- **Textbook:** *C for Business Programming*, by: John C. Molluzzo.
- **Class Time:** MWF 7:00–7:50 AM
- **Class Time:** MWF 2:00–2:50 PM
- **Final Exam Deadline:** Dec 3, 1999
- **Classroom:** GCB 150
- **Instructor (me):** Don Colton
- **My email:** don@colton.byuh.edu
- **My Office:** GCB 130 B, Phone: 293-3478
- **My Office Hours:** MWF 10–11
- **Teaching Assistant:** Margaret Miller
- **T.A. Hours:** MTWTh 6–10 PM, F 6–9 PM

1 Open Entry, Open Exit

This course is being structured into a new framework called "year round, open entry, open exit." Year round means offered all the time, even between semesters. Open entry means you can add the course at any time, even if the add-drop deadline is past. Open exit means you can totally complete the course (and know your final grade) ahead of schedule. You don't need to wait for the last day of class. It is an experimental concept.

We are still working on the "year round" and "open entry" parts of it, but "open exit" is working. You can take the final today if you like (but you only get one shot at it). Details below.

2 Why Take This Course?

In the old days (when I was young) IS professionals wrote programs. Today IS professionals (including you) must understand programming, even though many of them do not use it on a daily basis. For many, the focus of an IS professional's life has shifted from COBOL and RPG to the Internet. Often programs are bought off-the-shelf and customized rather than being built from scratch.

This however does not remove the need for an understanding of what goes on in a computer, or what goes into a program. I believe there will always be many IS jobs that require programming as a routine part of their workday, and people who can program will be sought-after and respected (and employed). (CGI scripting and automation of web pages come to mind.)

This course and its successor (IS 231) will teach you to program well enough that you can easily learn any language employers want, now or in the future. The foundation of most modern languages is ALGOL, and the most popular and respected language of that class is C. **You will learn C moderately well in this course, and much better in IS 231.** With the skill at C learned in this course, you will be able to continue learning C, or learn C++, JAVA, PERL, COBOL, RPG, BASIC, or any of the other languages (including 4GLs) that are likely to be encountered in IS settings. You will know the fundamentals of computer programming. After this one class, some of you that get "A"s will be good enough to get programming jobs.

Knowledge of operating systems is also very important. Today's client-side world is dominated by Microsoft Windows, but there is strong server-side pressure from Unix. UNIX and Windows are the two operating environments that I believe will dominate the IS computing world in the next decade and beyond. Therefore, this class will also introduce UNIX to a modest degree. You will learn the most commonly used commands, including those for file system maintenance (how to move from directory to directory, make new directories, move, rename, and delete files, etc.). You will learn to operate the most prominent free-software text editor, EMACS.

At the end of this course, you should feel comfortable listing C, UNIX, and EMACS among your skills on you résumé.

3 Prerequisites

The formal prerequisite is IS 190L (meaning you can use a computer to do word processing). I also expect you can add and subtract reliably in base 10 (normal numbers) without the aid of a calculator. I expect you can read and hear English (my English) with good comprehension.

I expect that you can manage your time well enough to get the work done. I expect that you can avoid the temptation to cheat. I expect that your handwriting is legible to me (although this is rarely required).

I assume you have **no experience** writing programs. We start from the very beginning in that regard. You must, however, be willing to work hard, two to three hours per class session.

4 Grading

Your grade is earned by getting points for completing labs and tests. When you have earned enough points, see me and I will certify your final grade. Once your is230 computer account is set up, progress reports are available to you by computer at any time.

dem programming labs	14 @ 10 pts	140
prob programming labs	3 @ 10 pts	30
pgm programming labs	32 @ 15 pts	480
quizzes (round 1)	16 @ 10 pts	160
quizzes (round 2)	4 @ 30 pts	120
final exam	1 @ 120 pts	120
total points possible		1050

930+	A	900–929	A-	870–899	B+
830–869	B	800–829	B-	770–799	C+
730–769	C	700–729	C-	670–699	D+
630–669	D	600–629	D-	0–599	F

Deadlines: Each assignment has a deadline. You can see these deadlines by sending email to GradeBot (see below) asking for a **status** report. The deadlines are "soft." Before the deadline an item is worth a certain number of points (100%). After the deadline, it is worth somewhat less each day until it reaches 60% of its original value. It then remains at the 60% level until the last day of class. All work must be completed by the end of the last day of class, except the final exam which may have a different deadline.

Incomplete and UW: If you quit working in the class before achieving a passing grade, I will probably give you a "UW" grade. In addition to saying that you failed the class, a UW also tells people that you didn't actually attempt the class; you just gave up.

I do not give "I" grades (incompletes) except in unusual circumstances. In my experience only a small fraction of incompletes are ever completed. I will consider giving you an incomplete if you request it, seem to have a good reason, have a pretty solid timeline for completion, and you get the necessary paperwork filled out.

5 Work Load (No Pain, No Gain)

The work load is about 30 hours reading, 60 hours programming, and 30 hours testing (preparing for and taking quizzes and tests). We spend about 40 hours in class (more on that below), and most of that time can be spent working on labs and preparing for quizzes.

Reading: The book is about 500 pages long. Here is my advice. Start out by skipping the reading in the chapter. Go directly to the lab assignment. Read it. If you can, just do it. If you can't, **then** go back and **skim** through the chapter. If you **still** can't do the lab assignment, go back and **carefully** read the chapter. You will probably find that the answers to most of your questions are given in the reading. (Since you have not programmed before, you will probably end up reading the whole book.)

Programming Labs: The key to this course is programming. That is the purpose of the class. That is your reason for being here. That is why you signed up. You want to learn to program. You will program. If you don't want to program, why are you taking this class?

You will write about fifty simple programs, and test and submit them for grading. Each program must run perfectly (more on that below) before it will be accepted. Most students will submit a program five or more times before it is accepted. The overall average time spent fixing and resubmitting programs apprears to be about 30 minutes per program.

Cheating: For some there will be a strong temptation to cheat by copying someone elses program. The rationalle is that since this class is so difficult, everybody else must be cheating too. Uh huh. This is not good for you. If you cheat in this way, what will you learn? You will learn that you cannot program. You will learn that you are a cheater. It will destroy your life. Count on it. Don't cheat. Just drop the class instead.

Tests: There are twenty quizzes and one final exam in this class. All of them (including the final) are given at the testing center using bubble sheets. You can complete the tests as soon as you want. I allow unlimited time and scratch paper, but no books, no notes, and no calculators. For each test, I will give you a sample test (with answers) that you can use as a study guide. You only get one chance to take each test. (If you feel there is some special reason you should get another chance, such as illness, discuss it with me.)

6 Lectures

Day One is an orientation to the class. During the next few class periods I demonstrate (using an overhead projector) how to do the lab work. This continues until most or all students have completed the first two labs.

After that, my theory of learning is that you know how to proceed. Indeed, I have discovered in past semesters that on average the students are so confident of their abilities that they ignore me and rush ahead into the assignments, with a goal to get enough points for an A so they can quit and think about something else. Good.

You will encounter problems that would be difficult to solve by yourself, so I and the TA are available to help you. But to get that help you must ask a question or state a request. For example, "Brother Colton, how do you do problem 17-2?" Each class period I will come to the class room and respond to these questions and requests.

Attendance: The calendar and the on-line **status** report give you a timeline for your progress through the tests and lab assignments. If you are making progress, you are counted present whether you actually come into the classroom or not. If you stop making progress, and have not earned a passing grade, you will be counted absent. If you have not been making progress and do not have a passing grade by the end of the semester (or term), you will receive a grade of "UW" (unofficial withdrawl) instead of an "F."

Due to INS (immigration) and VA (veterans) requirements the Vice President for Student Life is notified whenever a student misses four consecutive class days.

7 GradeBot (Yes Drill Sergeant Sir!)

GradeBot is my robotic program grader. It (he?) is available 24 hours a day, seven days a week, to grade and return your lab assignments. This is done via email. Obviously this means that you must maintain an email account to complete this class.

I provide you with a computer account on the is230 UNIX host. This account gives you access to a UNIX system, software (including compilers and assemblers), email, and some storage. Most of you will use this account to do all the lab work in this class. See me if you need any help getting set up.

GradeBot is correct and authoritative. It is your boss. It is your client. It is your Drill Sergeant. There is always a particular correct behavior that it wants. You must make your program behave in exactly the way that GradeBot is requiring. This may involve changing the wording of your prompts and/or the spacing and wording of your output. It will not significantly alter the difficulty of the problem.

To submit a program to GradeBot, send it by email to ¡gradebot@gradebot.byuh.edu¿. You can do this from almost anywhere on the Internet. The basic subject line for this class is "Subject: is230". That will get you a **status** report telling you everything you have completed, everything that is still due (and when), and what grade you have earned or are likely to get. To submit an assignment "x" to GradeBot, the subject line is "Subject: is230 x". If you are having problems with extra stuff appearing after your program (such as an advertisement for juno or hotmail), you can put a "BEGIN" line before your program and an "END" line after it. GradeBot does not understand attachments; your program must be in the body of your message. Do not use any special encoding, such as HTML or MIME.

It is possible but unlikely that GradeBot will make some major crazy mistake. If you find an example of this, bring it to me. I will generally reward you with some extra credit. Otherwise, you must assume GradeBot is right.

8 Lab Submission Rules

Cheating has occasionally been a problem in the past.

There are several rules that I use in this class. **These rules apply to the programs you submit to GradeBot.** They are designed to allow you to learn, but to prevent you from doing things that might let you pass the class without learning. Violation of any of these rules is typically regarded as a violation of the BYUH honor code. You will receive a score of zero for any such assignment, and it cannot be made up. Repeated violations may lead to failing the class. Please be careful what you submit.

The Keystroke Rule: Every keystroke in every lab you submit must come from **your own fingertips.** (If you are handicapped in some way that makes typing difficult or impossible for you, check with me. We can make a special exception for you if necessary.) You can re-use code that you wrote in a prior assignment (or in a prior class or in a prior job). You are forbidden to submit any code that was not typed by you yourself. You are permitted to copy things (particularly text strings) that GradeBot sends you in response to your submission.

The Open-Neighbor Rule: All labs are "open-neighbor" in the sense that you can **confer** with other students and lab assistants. You can read their code (if they let you). You can share your code with them. You can talk about your code, your approach, your difficulties, and your ideas. You can draw pictures and make analogies and ask the TA or me (even me) questions. You can use their ideas. However, you cannot submit their code to GradeBot, even if you first modify it.

The Collaboration Rule: You may collaborate with another person or a small group of people in working on programs. Typically this means you actively work out the solution together using one person's login account, and once the program is right, everybody else in the small group makes a copy and submits it. I require that every collaborative submission to GradeBot includes a comment near the top listing all the collaborators by name as joint authors. Otherwise your submission may be regarded as cheating.

The purpose of collaboration is **not** to merely get the work done. The purpose is to learn and understand. You must not submit as a collaboration anything that you do not personally understand.

It is impossible to collaborate on a program that is already finished. If you were not actively involved in creating the original work, you are **not** a collaborator.

The Looking Rule: Except for looking at the textbook, you are not allowed to look at your own code that you will be submitting and somebody else's code at the same time. If you look at somebody else's code, you must wait at least ten seconds before looking at your own code again, and vice versa. If you find yourself looking back and forth between your program and their program, you are probably copying, which is strictly against the rules.

The Challenge Rule: If I think that you may have violated these rules on some particular assignment, I will ask you (by email or in person) to state that you followed these rules. If I don't hear back from you, I will assume that you cheated and set your grade to zero for that assignment.

9 Types of Programming Labs

There are two types of programming labs: examples (named in the book demX-X or probX-X) and real work (named pgmX-X). Source code for examples is given right in the textbook. For real work you must invent it yourself.

dem Labs: The purpose of example labs is to encourage you to key in a fully-operational program and make it work. Why would it not work? There may be some small errors in the original program. Perhaps you will make a few typographical errors as you key it in. After submitting it for grading, you may want to "play" with the example program, changing various things to see what effect they have. In the end, you will learn good programming style and you should remember programming concepts better because you have worked through a detailed example.

On the example labs, you are permitted to submit a different program than the one shown in the textbook as long as it works the same. You can take this as a challenge to see if you can improve on the book version in various ways. Can you write the program in fewer lines? Can you organize it in a different way? But you can always fall back on the version in the book.

prob Labs: Prob labs are just like dem labs but they are generally longer and more complicated.

pgm Labs: The purpose of real-work labs is to experience programming and grow thereby. Programming can be an extreme joy, where time ceases to exist (e.g., hours pass quickly but you don't notice). It can be a great pleasure to cause a machine to produce reports and process data at your will. Or it can be a nightmare, where nothing seems to work right, and the most insignificant things turn out to have far too much significance, and you pull out great clumps of your hair and hit you head against the wall and you are glad that not every IS professional needs to be an accomplished programmer. Labs reflect the true reality of a programmer's life. You should experience labs.

10 Assignment Calendar

The dates on this list are not guaranteed. They are approximately correct. You should run a GradeBot status report to find the authoritative, correct due dates for you.

```
 1: 0101dem  thru Sep 03 (Fri) worth  10 pts
 2: q01      thru Sep 07 (Tue) worth  10 pts
 3: 0102dem  thru Sep 07 (Tue) worth  10 pts
 4: q02      thru Sep 08 (Wed) worth  10 pts
 5: 0106pgm  thru Sep 09 (Thu) worth  15 pts
 6: q03      thru Sep 09 (Thu) worth  10 pts
 7: 0201dem  thru Sep 09 (Thu) worth  10 pts
 8: q04      thru Sep 10 (Fri) worth  10 pts
 9: 0203pgm  thru Sep 11 (Sat) worth  15 pts
10: q05      thru Sep 13 (Mon) worth  10 pts
11: 0208pgm  thru Sep 14 (Tue) worth  15 pts
12: q06      thru Sep 15 (Wed) worth  10 pts
13: 0209pgm  thru Sep 17 (Fri) worth  15 pts
14: q07      thru Sep 18 (Sat) worth  10 pts
15: 0211pgm  thru Sep 20 (Mon) worth  15 pts
16: q08      thru Sep 20 (Mon) worth  10 pts
17: 0301dem  thru Sep 21 (Tue) worth  10 pts
18: q09      thru Sep 21 (Tue) worth  10 pts
19: 0301pgm  thru Sep 22 (Wed) worth  15 pts
20: q10      thru Sep 23 (Thu) worth  10 pts
21: 0302pgm  thru Sep 23 (Thu) worth  15 pts
22: q11      thru Sep 24 (Fri) worth  10 pts
23: 0307pgm  thru Sep 25 (Sat) worth  15 pts
24: q12      thru Sep 25 (Sat) worth  10 pts
25: 0309pgm  thru Sep 27 (Mon) worth  15 pts
26: q13      thru Sep 27 (Mon) worth  10 pts
27: 0407pgm  thru Sep 28 (Tue) worth  15 pts
28: q14      thru Sep 29 (Wed) worth  10 pts
29: 0501dem  thru Sep 29 (Wed) worth  10 pts
30: q15      thru Sep 30 (Thu) worth  10 pts
31: 0501prb  thru Oct 01 (Fri) worth  10 pts
32: q16      thru Oct 01 (Fri) worth  10 pts
33: 0501pgm  thru Oct 06 (Wed) worth  15 pts
34: q17      thru Oct 07 (Thu) worth  30 pts
35: 0503pgm  thru Oct 09 (Sat) worth  15 pts
36: q18      thru Oct 11 (Mon) worth  30 pts
37: 0509pgm  thru Oct 11 (Mon) worth  15 pts
38: q19      thru Oct 12 (Tue) worth  30 pts
39: 0601pgm  thru Oct 12 (Tue) worth  15 pts
40: q20      thru Oct 13 (Wed) worth  30 pts
41: 0606pgm  thru Oct 14 (Thu) worth  15 pts
42: 0612pgm  thru Oct 15 (Fri) worth  15 pts
43: 0703pgm  thru Oct 16 (Sat) worth  15 pts
44: 0707pgm  thru Oct 16 (Sat) worth  15 pts
45: 0708pgm  thru Oct 18 (Mon) worth  15 pts
46: 0801pgm  thru Oct 20 (Wed) worth  15 pts
```

```
47: 0807pgm  thru Oct 22 (Fri) worth  15 pts
48: 0809pgm  thru Oct 25 (Mon) worth  15 pts
49: 0901prb  thru Oct 26 (Tue) worth  10 pts
50: 0902dem  thru Oct 27 (Wed) worth  10 pts
51: 0903dem  thru Oct 27 (Wed) worth  10 pts
52: 0905dem  thru Oct 28 (Thu) worth  10 pts
53: 0909pgm  thru Oct 28 (Thu) worth  15 pts
54: 1001pgm  thru Oct 29 (Fri) worth  15 pts
55: 1001prb  thru Oct 30 (Sat) worth  10 pts
56: 1012pgm  thru Nov 02 (Tue) worth  15 pts
57: 1107dem  thru Nov 03 (Wed) worth  10 pts
58: 1105pgm  thru Nov 03 (Wed) worth  15 pts
59: 1108pgm  thru Nov 04 (Thu) worth  15 pts
60: 1205dem  thru Nov 05 (Fri) worth  10 pts
61: 1206dem  thru Nov 05 (Fri) worth  10 pts
62: 1212pgm  thru Nov 08 (Mon) worth  15 pts
63: 1305dem  thru Nov 08 (Mon) worth  10 pts
64: 1307dem  thru Nov 09 (Tue) worth  10 pts
65: 1312dem  thru Nov 09 (Tue) worth  10 pts
66: 1502pgm  thru Nov 11 (Thu) worth  15 pts
67: 1504pgm  thru Nov 12 (Fri) worth  15 pts
68: 1505pgm  thru Nov 15 (Mon) worth  15 pts
69: 1507pgm  thru Nov 17 (Wed) worth  15 pts
70: final    thru Dec 03 (Fri) worth 120 pts
```

11 Office Hours

Office hours are posted outside my office door. I also have an open-door policy, posted on my office door as follows: “If my door is open (even just a bit) feel free to knock and come in. – Bro. Colton” Students for whom the posted hours are not convenient, or who just want a guaranteed appointment, can contact me by email to make an appointment.

12 Course Evaluation

The course evaluation for IS 230 is given in the testing center. You will be required to complete the course evaluation right after you turn in quiz 20. (The testing center people will request you to do it then.) Your honest feedback (a) hopefully makes us feel good, (b) shows us where the course and/or instruction are weak, and (c) helps us improve the course for future students.

13 Subject to Change

It is very rare that I make major changes, but I might. If my changes seem unfair to you, let me know. I will try to fix it.

IS 230 Course Calendar — Fall 1999

Dates are approximate. See GradeBot Status for authoritative dates.

mtg	day	date	time	read	Topic	due (pts)
1	Wed	Aug 25	7am		Orientation, Syllabus	
2	Fri	Aug 27	7am		telnet, emacs, rmail	
3	Mon	Aug 30	7am	A	Computers, base 2, ascii, hex	
4	Wed	Sep 1	7am	1	Introduction, unix, gcc	
5	Fri	Sep 3	7am		unix, gcc	dem1-1, dem1-2
	Mon	Sep 6			**Holiday: Labor Day**	
6	Wed	Sep 8	7am	2	Integers	pgm1-6, dem2-1
7	Fri	Sep 10	7am		Integers	pgm2-3, pgm2-8
8	Mon	Sep 13	7am		Integers	pgm2-9, pgm2-11
9	Wed	Sep 15	7am	3	Real numbers	dem3-1
10	Fri	Sep 17	7am		Real numbers	pgm3-1, pgm3-2
11	Mon	Sep 20	7am		Real numbers	pgm3-7, pgm3-9
12	Wed	Sep 22	7am	4	Arithmetic operators	pgm4-7
13	Fri	Sep 24	7am	B	Program control, Structured programming	
14	Mon	Sep 27	7am	5	Indefinite iteration (while)	dem5-1
15	Wed	Sep 29	7am		Indefinite iteration	prob5-1, pgm5-1
16	Fri	Oct 1	7am		Indefinite iteration	pgm5-3, pgm5-9
17	Mon	Oct 4	7am	6	Definite iteration (for)	pgm6-1
18	Wed	Oct 6	7am		Definite iteration	pgm6-6, pgm6-12
19	Fri	Oct 8	7am	7	Program control (if)	pgm7-3
20	Mon	Oct 11	7am		Program control	pgm7-7, pgm7-8
21	Wed	Oct 13	7am	8	Complex decisions	pgm8-1
22	Fri	Oct 15	7am		Complex decisions	pgm8-7, pgm8-9
23	Mon	Oct 18	7am	9	Functions	dem9-2
24	Wed	Oct 20	7am		Functions	dem9-3, dem9-5
25	Fri	Oct 22	7am		Functions	prob9-1, pgm9-9
26	Mon	Oct 25	7am	10	Modular design (and testing)	prob10-1
27	Wed	Oct 27	7am		Modular design	pgm10-1, pgm10-12
28	Fri	Oct 29	7am	11	Arrays	dem11-7
29	Mon	Nov 1	7am		Arrays	pgm11-5, pgm11-8
30	Wed	Nov 3	7am	12	Pointers and strings	dem12-5
31	Fri	Nov 5	7am		Pointers and strings	dem12-6, pgm12-12
32	Mon	Nov 8	7am	13	Pointers, arrays, functions	dem13-5
33	Wed	Nov 10	7am		Pointers, arrays, functions	dem13-7, dem13-12
34	Fri	Nov 12	7am	15	File processing	
35	Mon	Nov 15	7am		File processing	pgm15-2
36	Wed	Nov 17	7am		File processing	pgm15-4, pgm15-5
37	Fri	Nov 19	7am		File processing	pgm15-7
38	Mon	Nov 22	7am		Review for Final	
39	Wed	Nov 24	7am	tba	tba/39	
	Thu	Nov 25			**Holiday: Thanksgiving Day**	
	Fri	Nov 26			**Holiday: Day after Thanksgiving**	
40	Mon	Nov 29	7am	tba	tba/40	
41	Wed	Dec 1	7am	tba	tba/41	
42	Fri	Dec 3	7am	tba	tba/42	