

IS 131 – Applications Program Development I

Course Syllabus and Calendar – Spring 1999

Instructor: Don Colton

Brigham Young University—Hawaii Campus

Abstract

- **Course Number:** IS 131
- **Title:** Applications Program Development 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. (Prerequisite: IS 190L.)
- **Textbook:** *C for Business Programming*, by: John C. Molluzzo. (Cost: \$58.55 at the BYUH bookstore, \$55.00 plus 3.95 shipping at Amazon.com)
- **Class Time:** MWF 7:00–8:50 AM
- **Final Exam Deadline:** Fri, Jun 18, 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:** Mon–Fri: 9:30pm–Midnight
- **T.A. Hours:** Tue, Thu: 3pm–6pm

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 this semester, we are starting “open exit.” 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 will be good enough to get 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 Linux, a free software version of 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 your 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 without the aid of a calculator. I expect you can read and hear English (my English) with good comprehension. I expect that you

can be on time to class and 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).

Most of 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 account is set up, progress reports are available 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	20 @ 10 pts	200
final exam	1 @ 200 pts	200
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

5 Work Load

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, and most of that can be spent working on labs and preparing for quizzes.

Reading: The book is about 500 pages long. Skip the chapter and 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 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 is about 30 minutes per program.

Cheating: For some there will be a strong temptation to cheat by copying someone else's program. This is not good for you. If you cheat in this way, what will you learn? It will destroy your life. Count on it.

Tests: There are about twenty tests in this class. All of them (including the final) are given at the testing center. You can complete the tests as soon as you want. Generally I allow unlimited time but no books, notes, or calculators. For each test, I will give you a sample test (generally 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 Deadlines

Each lab and test has a deadline. You can see these deadlines by sending email to GradeBot 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 one point less per 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.

Status Report: You can ask GradeBot for a status report at any time. It tells you all the deadlines that currently apply to you, and what grade you have earned to that point.

7 Lectures

Each class begins with a short lecture. After that, class time is dedicated answering your questions, reviewing for tests, or doing lab work. The short lecture introduces new material or discusses a test or lab that is due soon. During the lecture your attention is requested. Computers should not be used until I say you can go ahead and work.

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.

Attendance: The calendar and the status report give you a timeline for your progress through the tests and lab assignments. If you are ahead of schedule, you are counted present whether you actually attend or not. If you are behind schedule on the labs you must attend class or you will be counted absent.

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

8 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.

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. Likely this will 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.

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.

9 Lab Submission Rules

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 a violation of the 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.

9.1 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.

9.2 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.

9.3 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.

9.4 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.

10 Types of Programming Labs

There are two types of programming labs: examples (named 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? 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 probably 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 your 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.

11 Office Hours

Office hours are posted outside my office door. Currently they are Daily 1–2. Office hours are subject to change, as I might discover the need to attend some meeting somewhere, or visit the men's room, or talk to someone in the computer lab.

Students for whom the posted hours are not convenient, or who just want a guaranteed appointment, can come by whenever my door is open (which is most of the time) or contact me by email to make an appointment.

My “open-door policy” is 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”

12 Communication by Email

When I want to say something, or when you want to say something, if we are not in the same room, my first choice is to do it by email. I far prefer it to telephone calls, for instance. When there is an announcement, I will generally tell you in class or send it to you by email. Such announcements might include clarifications on the lab assignments. You must maintain an email account and to provide me with a valid email address.

13 Computer Accounts

You will have a computer account on the is131 UNIX host. This account gives you access to UNIX systems, software (including compilers and assemblers), email, and limited storage. Most of you will use this account to do all the lab work in this class. See the teaching assistant if you need any help getting set up.

Programming homework is submitted by email from the student to me by sending it to “GradeBot,” who grades and returns such homework to the sender by email. You will probably use your is131 account for this. However, you can send from your byuh account or from hotmail or from road runner or lava net or aol or wherever.

14 Assignment Calendar

1:	0101dem	thru	May 03	(Mon)	worth	10	pts
2:	q01	thru	May 03	(Mon)	worth	10	pts
3:	0102dem	thru	May 04	(Tue)	worth	10	pts
4:	q02	thru	May 04	(Tue)	worth	10	pts
5:	0106pgm	thru	May 05	(Wed)	worth	15	pts
6:	q03	thru	May 05	(Wed)	worth	10	pts
7:	0201dem	thru	May 05	(Wed)	worth	10	pts
8:	q04	thru	May 06	(Thu)	worth	10	pts
9:	0203pgm	thru	May 06	(Thu)	worth	15	pts
10:	q05	thru	May 06	(Thu)	worth	10	pts
11:	0208pgm	thru	May 08	(Sat)	worth	15	pts
12:	q06	thru	May 08	(Sat)	worth	10	pts
13:	0209pgm	thru	May 10	(Mon)	worth	15	pts
14:	q07	thru	May 10	(Mon)	worth	10	pts
15:	0211pgm	thru	May 10	(Mon)	worth	15	pts
16:	q08	thru	May 11	(Tue)	worth	10	pts
17:	0301dem	thru	May 11	(Tue)	worth	10	pts
18:	q09	thru	May 11	(Tue)	worth	10	pts
19:	0301pgm	thru	May 12	(Wed)	worth	15	pts
20:	q10	thru	May 13	(Thu)	worth	10	pts

21:	0302pgm	thru	May 13	(Thu)	worth	15	pts
22:	q11	thru	May 13	(Thu)	worth	10	pts
23:	0307pgm	thru	May 14	(Fri)	worth	15	pts
24:	q12	thru	May 14	(Fri)	worth	10	pts
25:	0309pgm	thru	May 14	(Fri)	worth	15	pts
26:	q13	thru	May 15	(Sat)	worth	10	pts
27:	0407pgm	thru	May 15	(Sat)	worth	15	pts
28:	q14	thru	May 17	(Mon)	worth	10	pts
29:	0501dem	thru	May 17	(Mon)	worth	10	pts
30:	q15	thru	May 17	(Mon)	worth	10	pts
31:	0501prb	thru	May 17	(Mon)	worth	10	pts
32:	q16	thru	May 17	(Mon)	worth	10	pts
33:	0501pgm	thru	May 19	(Wed)	worth	15	pts
34:	q17	thru	May 19	(Wed)	worth	10	pts
35:	0503pgm	thru	May 21	(Fri)	worth	15	pts
36:	q18	thru	May 21	(Fri)	worth	10	pts
37:	0509pgm	thru	May 22	(Sat)	worth	15	pts
38:	q19	thru	May 22	(Sat)	worth	10	pts
39:	0601pgm	thru	May 22	(Sat)	worth	15	pts
40:	q20	thru	May 24	(Mon)	worth	10	pts
41:	0606pgm	thru	May 24	(Mon)	worth	15	pts
42:	0612pgm	thru	May 24	(Mon)	worth	15	pts
43:	0703pgm	thru	May 25	(Tue)	worth	15	pts
44:	0707pgm	thru	May 25	(Tue)	worth	15	pts
45:	0708pgm	thru	May 25	(Tue)	worth	15	pts
46:	0801pgm	thru	May 27	(Thu)	worth	15	pts
47:	0807pgm	thru	May 28	(Fri)	worth	15	pts
48:	0809pgm	thru	May 29	(Sat)	worth	15	pts
49:	0901prb	thru	May 31	(Mon)	worth	10	pts
50:	0902dem	thru	May 31	(Mon)	worth	10	pts
51:	0903dem	thru	May 31	(Mon)	worth	10	pts
52:	0905dem	thru	May 31	(Mon)	worth	10	pts
53:	0909pgm	thru	May 31	(Mon)	worth	15	pts
54:	1001pgm	thru	Jun 01	(Tue)	worth	15	pts
55:	1001prb	thru	Jun 01	(Tue)	worth	10	pts
56:	1012pgm	thru	Jun 03	(Thu)	worth	15	pts
57:	1107dem	thru	Jun 03	(Thu)	worth	10	pts
58:	1105pgm	thru	Jun 04	(Fri)	worth	15	pts
59:	1108pgm	thru	Jun 04	(Fri)	worth	15	pts
60:	1205dem	thru	Jun 04	(Fri)	worth	10	pts
61:	1206dem	thru	Jun 05	(Sat)	worth	10	pts
62:	1212pgm	thru	Jun 07	(Mon)	worth	15	pts
63:	1305dem	thru	Jun 07	(Mon)	worth	10	pts
64:	1307dem	thru	Jun 07	(Mon)	worth	10	pts
65:	1312dem	thru	Jun 07	(Mon)	worth	10	pts
66:	1502pgm	thru	Jun 08	(Tue)	worth	15	pts
67:	1504pgm	thru	Jun 09	(Wed)	worth	15	pts
68:	1505pgm	thru	Jun 10	(Thu)	worth	15	pts
69:	1507pgm	thru	Jun 11	(Fri)	worth	15	pts
70:	final	thru	Jun 18	(Fri)	worth	200	pts

15 Subject to Change

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

IS 131 Course Calendar — Spring 1999

All due dates are approximate. See GradeBot Status for authoritative dates.

mtg	day	date	time	read	Topic	due (pts)
1	Wed	Apr 28	7am		Orientation, Syllabus	
2	Wed	Apr 28	8am		telnet, emacs, rmail	
3	Fri	Apr 30	7am	A	Computers, base 2, ascii, hex	
4	Fri	Apr 30	8am	1	Introduction, unix, gcc	
5	Mon	May 3	7am		unix, gcc	dem1-1, dem1-2
6	Mon	May 3	8am	2	Integers	pgm1-6, dem2-1
7	Wed	May 5	7am		Integers	pgm2-3, pgm2-8
8	Wed	May 5	8am		Integers	pgm2-9, pgm2-11
9	Fri	May 7	7am	3	Real numbers	dem3-1
10	Fri	May 7	8am		Real numbers	pgm3-1, pgm3-2
11	Mon	May 10	7am		Real numbers	pgm3-7, pgm3-9
12	Mon	May 10	8am	4	Arithmetic operators	pgm4-7
13	Wed	May 12	7am	B	Program control, Structured programming	
14	Wed	May 12	8am	5	Indefinite iteration (while)	dem5-1
15	Fri	May 14	7am		Indefinite iteration	prob5-1, pgm5-1
16	Fri	May 14	8am		Indefinite iteration	pgm5-3, pgm5-9
17	Mon	May 17	7am	6	Definite iteration (for)	pgm6-1
18	Mon	May 17	8am		Definite iteration	pgm6-6, pgm6-12
19	Wed	May 19	7am	7	Program control (if)	pgm7-3
20	Wed	May 19	8am		Program control	pgm7-7, pgm7-8
21	Fri	May 21	7am	8	Complex decisions	pgm8-1
22	Fri	May 21	8am		Complex decisions	pgm8-7, pgm8-9
23	Mon	May 24	7am	9	Functions	dem9-2
24	Mon	May 24	8am		Functions	dem9-3, dem9-5
25	Wed	May 26	7am		Functions	prob9-1, pgm9-9
26	Wed	May 26	8am	10	Modular design (and testing)	prob10-1
27	Fri	May 28	7am		Modular design	pgm10-1, pgm10-12
28	Fri	May 28	8am	11	Arrays	dem11-7
	Mon	May 31			Holiday: Memorial Day	
29	Wed	Jun 2	7am		Arrays	pgm11-5, pgm11-8
30	Wed	Jun 2	8am	12	Pointers and strings	dem12-5
31	Fri	Jun 4	7am		Pointers and strings	dem12-6, pgm12-12
32	Fri	Jun 4	8am	13	Pointers, arrays, functions	dem13-5
33	Mon	Jun 7	7am		Pointers, arrays, functions	dem13-7, dem13-12
34	Mon	Jun 7	8am	15	File processing	
35	Wed	Jun 9	7am		File processing	pgm15-2
36	Wed	Jun 9	8am		File processing	pgm15-4, pgm15-5
37	Fri	Jun 11	7am		File processing	pgm15-7
38	Fri	Jun 11	8am		Review for Final	
39	Mon	Jun 14	7am	tba	tba/39	
40	Mon	Jun 14	8am	tba	tba/40	
41	Wed	Jun 16	7am	tba	tba/41	
42	Wed	Jun 16	8am	tba	tba/42	
	Fri	Jun 18			Final Exam in testing center	Final Exam (200)
	Mon	Jun 21	noon		Final Grades by email (probably)	