IS 231 – Computer Programming II
Course Syllabus and Calendar – Spring 2001

Professor Don Colton

Brigham Young University—Hawaii Campus

1 Brief Overview

Companies everywhere are eager to get onto the World Wide Web. They can choose to be present (but not special) with off-the-shelf software, or they can grab for market share with a more exciting and satisfying user experience, using home-grown or customized software. Programmers have become as vital as accountants in this new world order.

This course and its predecessor (IS 230) teach you to program well enough that you can easily learn any language employers want, now or in the future. Perl (which is in the same language family as C, C++, and Java) is probably the most popular language on the server side of the web.

1.1 The Course

- **Course Number:** IS 231
- **Title:** Computer Programming II
- **Updated Course Description:** Emphasis on web programming (CGI, sockets), problem solving, stacks, queues, associative arrays, regular expressions, data manipulation, and simple algorithm analysis. Review of looping and precedence. (Prerequisite: IS 230 or equivalent.)
- **Required Textbook:** *Learning Perl*, by: Randal Schwartz and Tom Christiansen. ISBN 1-56592-284-0
- **Recommended Textbook:** *Programming Perl*, by: Larry Wall, Tom Christiansen, and Randal Schwartz. ISBN 1-56592-149-6
- **Class Time:** TTh 7:00–9:50 AM
- **Final Exam:** Thu 14 Jun, 7:00–9:50 AM
- **Classroom:** GCB 140

1.2 The Instructor

- **Instructor (me):** Don Colton
- **My email:** don@colton.byuh.edu
- **My Office:** GCB 130 B
- **T.A. Hours:** Mon–Fri, 8–10 PM
- **T.A. Location:** GCB 140

1.3 Grading

Your grade is based on points. 1000 assigned points are assigned, and some extra credit points are also available. The grading will be as follows:

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

To get above a C, you must also pass the final exam (ten programming problems, ten points each) with a sufficient score, as follows:

93+	A	90–92	A-	87–89	B+
83–86	B	80–82	B-	77–79	C+

Your final grade will be the **lower** of these two grades: your total points grade and your final exam grade.

Grading is discussed further below.

1.4 Office Hours

Office hours are currently MWF 1–2 and TTh 10–11 (except devotionals and such things). Updated office hours are posted outside my office door. Students for whom the posted hours are not convenient can contact me by email to make an appointment.

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"

1.5 Students with Special Needs

If you require accommodation for special learning needs or physical impairments, please see me as soon as possible.

1.6 Subject to Change

I am revising some aspects of the course at this point, but any changes I make are likely to be to your advantage. If any of my changes seems unfair to you, let me know. I will try to correct it.

2 Now, About the Course

I assume that you want a programming job, or at least the ability to use programming in your future job. Some of you may be ready right now. At the successful conclusion of this course, many of you will be qualified to start work in most entry-level IS programming jobs.

We will develop your programming skills by completing projects in areas that support electronic commerce on the web. We will develop your knowledge of a number of additional topics that you are likely to encounter in programming.

Knowledge of operating systems is also very important. Today's client-side world seems dominated by Microsoft Windows, but there is a strong server-side presence 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 utilizes UNIX to a modest degree. You will know 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 know how to operate the most prominent free-software text editor, EMACS.

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

2.1 What is the Course Like?

Much like IS 230, you will write programs that are graded by my robotic grader, GradeBot. Class time will be devoted to understanding basic concepts of computer programming as applied to the World Wide Web.

2.2 Prerequisites

The prerequisite is IS 230 (Computer Programming I). I assume that you have written some programs. You know how to do formatted printing, and use **if**, **else**, **while**, **do while**, **for**, and subroutines. I assume that you have some skill, but you are not ready to sit in an interview and claim that you are a programmer.

3 Grading

Your grade is earned by getting points for completing labs, readings, and tests. Once your computer account is set up, progress reports are available to you by computer at any time.

att	attendance	100 pts
pgm	programming labs	500 pts
qic	in-class quizzes/final	140 pts
qtc	testing center quizzes	160 pts
read	readings	100 pts
tot	total points possible *	1000 pts

* Extra credit is also available.

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

The in-class quizzes and the final exam are programming tests. You will be asked to write several fairly simple programs. The tests must be taken in class at the designated time. This is because anyone with advance knowledge of the test could memorize answers easily, so I must make sure no one has advance knowledge. If you must take the test at some other time, I must create a totally separate test for you.

Incomplete and UW: If you quit working in the class before achieving a passing grade, I will probably give you a "UW" grade instead of an "F."

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.

4 Work (No Pain, No Gain)

Most of your time will be spent writing programs. I am not sure how much time it would take a good student programmer to complete all of these assignments. A professional could probably do all of them in a week. But you are not a professional yet. The work is difficult mainly because it is unfamiliar. Our task is to make it familiar, and therefore easier. You will find that assignments you did in three or four hours early in the semester can be done much more quickly later in the semester. You should feel a great sense of achievement.

Reading: Perl is a serious and powerful language. *Learning Perl* by Schwartz and Christiansen is a light book that covers the most-used features of the Perl language. It includes a chapter on CGI programming. The

book seems to me very easy to read. If your programming background is shallow, you will probably want to begin with this book.

Programming Perl by Wall, Christiansen, and Schwartz is a much more serious and detailed treatment of the Perl Language. Larry Wall is the primary creator of the language, so this book is very authoritative and much more complete. It is also quite well written. If your programming skills are more advanced, you may want to skip the "Learning" book and go straight to this "Programming" one.

I do give credit for reading in the books. To honestly get reading credit, you must let your sight rest on each of the words in the assignment, and you must try to understand what is being said. If you can speed-read some or all of it with reasonable comprehension, that is acceptable too. Of course, that is not really adequate for understanding, and you should strive for understanding as well.

As you complete each assigned reading mentioned on the course calendar, notify me by submitting a program that tells me "I read chapter 1, Introduction, of Learning Perl by Schwartz and Christiansen." Or something like that. Email GradeBot (see below) for authoritative details. When you submit such a program, you are asserting that you have in fact done what the program says about you.

Labs: The key to a programming course is programming. (Duh.) You will complete a number of programs. Programs are graded by GradeBot (see below). Each must run perfectly before it will be accepted.

Tests: There are six tests given in the testing center using bubble sheets, including one that is comprehensive and is due the last day of class. 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, a sample test is available through GradeBot for you to 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.)

There are three tests given in class. These are programming tests where you will be asked to write some particular program without the aid of notes of a computer: written by hand, graded by hand. Time is limited. Scratch paper is provided. No books, no notes, and no calculators. No sample tests are provided.

5   Lectures

Those who took IS 230 from me will wonder whether IS 231 will be the same. As you will recall, in that class each day I asked "Do you have any questions?" and then simply (or elaborately) answered them. Because students moved at vastly different paces, based on their vastly different backgrounds and aptitudes, the discussion was not always meaningful to everyone. Some were behind. Some were ahead. Some stopped attending because they got so far ahead. That was okay.

In IS 231 the differences is much less. We are all "up to speed." We have more that we can talk about together. It is more likely that the discussion will be meaningful to nearly everyone.

I still like that general "got questions?" philosophy. It leaves the responsibility for learning with the people that are supposed to learn: the students. (I cannot learn for you.) Canned lectures can be fun and exciting, but often I have found myself simply presenting material that was already presented in the reading, which discouraged students from even doing the reading. This did not seem the best use of limited class time.

You do your part by reading, attempting the labs, deciding what questions to ask in class, and bravely asking them. I prepare the overall calendar, the syllabus, the list of assignments, the GradeBot routines to grade them, and the schedule of readings. I also prepare myself to answer whatever questions you can think of.

Attendance: I take roll in this class, and attendance counts for 10% of your final grade. Generally each day is worth 8 points. I take 8-point roll at the start of class. I take 7-point roll about 15 minutes into class. I take 6-point roll about 30 minutes into class. I take 5-point roll at the end of class. Missing persons get zeros.

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

6   GradeBot

GradeBot is my robotic program grader. It is generally available 24 hours a day, seven days a week, to grade and return your lab assignments. This is currently done via email.

I enable you to have a computer account on the (is230.byuh.edu) Linux host. This account gives you access to a UNIX system, software (including compilers and assemblers), email, and some storage (type "quota" to see how much). You can also put up your own web page and cgi scripts. 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.

For grading, 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 demands. You must make your program behave in exactly the way that GradeBot is requiring (including spelling errors, if any). Be sure to look at a sample "conver-

sation" with GradeBot before you start writing your program.

To submit a program to GradeBot, send it by email to <gradebot@is230.byuh.edu> (or just gradebot if you are already logged in there). You can do this from almost anywhere on the Internet. The basic subject line for this class is "Subject: is231". With "Subject: is231 status" you get 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 "xxx" to GradeBot, the subject line is "Subject: is231 xxx". 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. Currently 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.

If you discover a case where you believe that Grade-Bot is wrong, tell me about it. If you found an error in GradeBot, I generally reward you with some extra credit. Otherwise, you must assume GradeBot is right.

7   Lab Submission Rules

Cheating has never been a problem in this class, but there are rules. I am unhappy when I see cheating in any class. These may be cases where one student gives a copy of their completed program to another student, and the second student keys it in, possibly with minor changes, such as changing the names of variables. In worse cases, the second student uses cut-and-paste to copy the program, or sections of it. In almost every case, the second student *does not understand how the program works,* or why the program says what it says. I consider any such behavior to be plagiarism and an honor code violation. I want you to learn, but not do things that might let you complete the assignments without learning.

Open-Neighbor versus Copying: All labs are "open-neighbor" in the sense that you can *confer* with other people. You can read their code (if they let you). You can show your code to them. You can talk about your code, your approach, your difficulties, and your ideas. You can draw pictures and make analogies and ask questions. You can use their ideas. However, *you cannot make a copy of their code or submit their code to GradeBot, even if you first modify it.*

Never let another student take, borrow, or keep a copy of any program you wrote for this class. You can look at it *together.* If it is printed, please look at it away from any computers. If it is online, look at it on the author's own screen. Never bring up a window on the second student's screen so they can look at the first student's program. You can talk about what the program does, and why it is that way. Do NOT leave them with a copy of your program.

If you receive a copy of a program from someone, and use it as the basis for the program you are submitting, you are cheating.

8   Programming Labs

The purpose of lab work 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.

9   Course Calendar

Generally the lectures and discussion in class will follow the due dates for the various assignments (shown below). In-class tests will generally occur at the start of class.

10   Due Dates and Points

The names, dates, and points on this list are not guaranteed, but they are approximately correct. You should run a GradeBot status report to find the authoritative, correct due dates for you. The wording in this list is condensed to make it fit the space available.

```
    1: hello      thru Apr 26 (Thu)   10 pts
    2: sc1        thru Apr 27 (Fri)   15 pts
    3: sc2        thru Apr 28 (Sat)   10 pts
    4: sc3        thru Apr 30 (Mon)   10 pts
    5: a02        thru May 01 (Tue)    4 pts
    6: sumProd    thru May 01 (Tue)   20 pts
    7: qprec      thru May 02 (Wed)   20 pts
    8: a03        thru May 03 (Thu)    8 pts
    9: sc5        thru May 03 (Thu)    5 pts
   10: starbox    thru May 03 (Thu)   20 pts
   11: sc7        thru May 05 (Sat)   10 pts
   12: etut       thru May 07 (Mon)   20 pts
   13: a04        thru May 08 (Tue)    8 pts
   14: qregex     thru May 08 (Tue)   20 pts
   15: roman      thru May 08 (Tue)   25 pts
   16: sc4        thru May 09 (Wed)    5 pts
   17: a05        thru May 10 (Thu)    8 pts
   18: sc8        thru May 10 (Thu)    5 pts
   19: sc9        thru May 11 (Fri)    5 pts
   20: qloops     thru May 12 (Sat)   20 pts
   21: sc19       thru May 14 (Mon)   15 pts
   22: a06        thru May 15 (Tue)    8 pts
   23: cgi1       thru May 15 (Tue)   25 pts
   24: qic1       thru May 15 (Tue)   20 pts
   25: cgi2       thru May 16 (Wed)   25 pts
   26: qbigoh     thru May 16 (Wed)   20 pts
   27: a07        thru May 17 (Thu)    8 pts
   28: hfetch1    thru May 18 (Fri)   25 pts
   29: hfetch2    thru May 21 (Mon)   25 pts
   30: a08        thru May 22 (Tue)    8 pts
   31: checkwr1   thru May 22 (Tue)   25 pts
   32: qprintf    thru May 22 (Tue)   20 pts
   33: checkwr2   thru May 23 (Wed)   40 pts
   34: a09        thru May 24 (Thu)    8 pts
   35: href1      thru May 25 (Fri)   30 pts
   36: href2      thru May 26 (Sat)   30 pts
   37: a10        thru May 29 (Tue)    8 pts
   38: qic2       thru May 29 (Tue)   20 pts
   39: href3      thru May 30 (Wed)   30 pts
   40: a11        thru May 31 (Thu)    8 pts
   41: sitemap1   thru Jun 01 (Fri)   30 pts
   42: sitemap2   thru Jun 02 (Sat)   30 pts
   43: sitemap3   thru Jun 04 (Mon)   50 pts
   44: a12        thru Jun 05 (Tue)    8 pts
   45: index      thru Jun 06 (Wed)   30 pts
   46: a13        thru Jun 07 (Thu)    8 pts
   47: preced     thru Jun 09 (Sat)   30 pts
   48: a14        thru Jun 12 (Tue)    8 pts
 * 49: nkrypto    thru Jun 12 (Tue)   20 pts
```

```
 * 50: robot      thru Jun 12 (Tue)   40 pts
 * 51: wcs1       thru Jun 12 (Tue)   15 pts
 * 52: zoo1       thru Jun 12 (Tue)   40 pts
 * 53: zoo2       thru Jun 12 (Tue)   50 pts
   54: final-tc   thru Jun 14 (Thu)   60 pts
   55: final-ic   thru Jun 14 (Thu)  100 pts
```