

CS 201 – Web Programming

Course Syllabus and Calendar – Spring 2007

Professor Don Colton

Brigham Young University Hawaii

The Course

- **Course Number:** CS 201
- **Title:** Web Programming
- **Course Description:** Review of CS 101. Problem solving, stacks, queues, hash tables, mathematical analysis of algorithms, regular expressions. Web programming including CGI and database. (Prerequisites: CS 101 and Math 106 or higher.)
- **Textbook (highly recommended):** *Perl by Example*, by: Ellie Quigley. ISBN 0-13-028251. SRP new \$44.99. Used about \$35.
- **Class Time:** MWF 10:00–10:50 AM
Final Exam: Fri 20 Apr, 11:00–2:00 PM
Classroom: GCB 140

The Instructor

- **Instructor (me):** Don Colton
- **My email:** don@colton.byuh.edu
- **My Office:** GCB 130 B
- **Office Hours:** Daily 11 AM to 11:50 AM

Open-Door Policy

My office hours are shown above. You can contact me by email to make an appointment at another time. I also have an open-door policy: **If my door is open (even just a bit) feel free to knock and come in.**

The CS Tutors

- check at GCB 101 for changes
- **T.A. Location:** GCB 101 (CS Lab)
- **T.A. Hours:** Mon–Thu, 3 PM to 11:30 PM **
- ** May stay open til 1 AM based on demand.
- **T.A. Hours:** Fri, 3 PM to 8 PM
- **T.A. Hours:** Sat, 3 PM to 6 PM

1 Course Overview

By the end of this semester, you will be writing useful web programs using databases. A web program is one that displays information on your web browser and lets you fill in the blanks or press buttons to make requests. A database lets you store and retrieve a large amount of information.

For example, you could write a web program to key in your CD collection and later key in a song title and have the computer tell you which CD it is on. But you probably have them downloaded to your MP3 player already.

Or you could make a database for a used textbook business, where you key in title, price, owner, and phone number. Maybe you could collect a commission from every sale.

In this class you will learn valuable technical skills to help you become the master of the computers in your life. You will learn to write useful (not just toy) programs. You will find out what it takes. Maybe you will love it and go on to be a world-class programmer. Maybe you will write an occasional program to solve a need. Maybe you will just ask someone else, but you will know what to ask and expect.

This class teaches you to write programs in Perl for the Web using SQL databases at an intermediate level of programming proficiency.

Perl is a popular scripting language. It is very powerful, has been around for a long time, and is broadly available. It is even free. Amazon.com, one of the most famous web sites in history, was originally written in perl. (For speed, many parts of amazon.com have been rewritten in faster languages.) Perl makes things easy for the programmer, at the expense of being slower than C and C++.

The **Web** means your favorite browser, such as Microsoft Internet Explorer, Mozilla Firefox, Opera, or even Lynx. You will create web pages dynamically, using forms, checkboxes, radio buttons, fill-in

blanks, and “submit” buttons. You will respond to the data entered into them. We will use CGI, the common gateway interface standard, as our processing method.

SQL, structured query language, is the international standard for database access. You will be introduced to the basic commands for storing data, retrieving data, and updating data. We will use MySQL, a powerful and inexpensive implementation of SQL, as our database tool.

It’s all free. We will not tie you down to any proprietary and expensive languages and interfaces. Instead we will learn things that you can take with you right now and use on your first job without needing the chief financial officer to write a big check to get you started. You can run these tools at home or generally any place in the world.

But what if your next job requires you to use Microsoft SQL? Or Microsoft ASP? No problem. The tools and design methods you learn will prepare you to understand and work with specialized tools that may be in use elsewhere. Those of you that drive “stick” will realize that it is easy to drive a car with an automatic transmission. Those of you that only drive an “automatic” may find it is difficult to drive “stick.” That is why we are teaching you to drive “stick” in this class.

Programming proficiency means that you can get something done without a lot of help. If you always need help for everything, you are not proficient, and you are a drag on your organization. You will need to be a contributor. In this class you need to be able to write decent code without help on the midterms and final exam.

By the time you finish this course, you will be programming web pages. Not just HTML, but Forms and server-side CGI, relational data base entry, maybe even games. You will have some marketable skills. If you are a Math major or an Information Systems major, completion of this course constitutes “programming proficiency” for your major. IS and Math are not all about programming, but those without programming skills are at a substantial disadvantage in the job market. For Math Ed, a little programming skill can let you develop custom math-based learning games on the computer.

For **Computer Science** majors, or other students who want more than an intermediate level of programming proficiency, there is yet more to learn. This course is just a stepping stone.

Prerequisites: The formally designated prerequisite is CS 101 (Beginning Computer Programming). In this case, I assume that you have written some programs. You probably know how to do formatted printing. You can use **if, else, while, do while, for**, subroutines, and arrays. You are no longer clueless, but you are not yet confident.

Alternately, you can take the class if you have already completed some Calculus. Your level of mathematical maturity should allow you to learn the prerequisite material during our review (the first third of the course).

If you do not have prior Calculus or programming experience, you should confer with the instructor to make sure this is the right class for you.

2 Web Hosting

Today’s Internet provides individuals and businesses with several major options for getting web content up and running. Large organizations (like BYUH) often build and operate their own data centers. With such an arrangement, you can have physical access to your computers, and sit directly at the console to control the configuration and programming.

Smaller organizations often purchase a web hosting plan from a web hosting provider. The actual servers are typically located in data centers far away from you, the customer. Many of these plans cost less than \$10 per month. With such an arrangement, you receive a set amount of space (say 100 gigabytes) and a set amount of bandwidth (say 100 gigabytes per month). Your access to these resources is generally through the web, or by FTP, or by SSH.

With remote hosting, configuration of your account is typically done using a control panel. The control panel is a web interface that allows you to add and delete email accounts and mailing lists, databases, ftp users, domain names, and other things. The control panel also provides access to reports, such as disk space used and bandwidth used.

cPanel (<http://www.cpanel.com/>) and H-Sphere (<http://www.hsphere.com/>) are the major control panels in use today.

Your “is2” Account

To provide a realistic hosting environment, I have adopted the cPanel web hosting environment. I have elected to give each student their own web account, subject to a few rules. The account I provide is

similar in features to what you could purchase from a web hosting provider for a few dollars per month.

See <http://is2.byuh.edu/> for rules that apply to your account. Here are some of the rules.

(1) You can use your web space both for school assignments and also for personal things, such as personal or family web pages, blogs, and photo albums. You can even create personal email addresses.

(2) You will have access to your account through a control panel (currently cPanel). You can upload files through FTP and SSH. You can get command-line access through SSH.

(3) We will create for you a personal domain name, something like `xxxx.is2.byuh.edu`, where `xxxx` is replaced by a name of your choice not already taken by someone else. In addition, if you wish, you can purchase a personal domain name from a domain registrar. Typically this costs less than \$10 per year.

(4) We intend that you can keep this free account forever, or at least for an extended amount of time, even after you graduate. This will depend on our experience as things go along, and could change at any time.

(5) You are required to keep your contact email address up to date. This should be easy for you to do. We contact you every month at that address. If you stop responding, we will assume you have abandoned your account and we will close it.

3 Grading

Usually A is the most common grade in this class, but for some students, this is a really hard class. Every semester there are a few retakes. The chart below shows the number of students earning each grade recently.

semester	A	B	C	D	F	UW
2004 Win	16	10	9	4	9	.
2004 Spr	5	4	7	5	.	2
2004 Fall	9	5	12	6	.	1
2005 Win	13	4	10	8	5	1
2005 Spr	6	5	3	8	1	.
2005 Fall	5	4	2	3	.	.
2006 Win	19	2	4	9	2	1
2006 Spr	17	7	8	6	.	.
2006 Fall	10	1	8	11	1	2
2007 Win	8	6	8	4	1	1

Your total-points grade is based on 1000 points of assigned work. Some extra credit points may be available. The total-points grading is 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

Normally the points available are as follows. We include some “safety net” activities such as in-class practices. There is a penalty for late work.

pgm	programming labs	400 (450) pts
prac	in-class practices	125 (140) pts
qtc	testing center quizzes	175 (200) pts
qic	midterm exams	150 pts
final	final exam (in class)	150 pts
tot	total points assigned	1000 pts

The notation “400 (450) pts” means that up to 400 points go toward your final grade, but 450 points are actually available to be attempted. The extra points are a “life line.” It is a bit like extra credit. You can miss up to 50 points in that category and still get full credit toward your final grade. If you have more than 400 points, then you have a perfect score in that area. The extra points do not carry over into other areas.

Genius Method: Some people are so smart that they don’t need to come to class. At your request, I will compute your grade this way instead. The in-class practice points are redistributed.

pgm	programming labs	500 (565) pts
qic	in-class quizzes	150 pts
qtc	testing center quizzes	100 (111) pts
final	final exam (in class)	250 pts
tot	total points assigned	1000 pts

The Final Exam: This is the part that gets some people into trouble. I have been compelled to add this rule because of occasional students who do well on points but cannot perform at the end of the semester. I believe that a grade of B or A means you can perform well on demand.

Therefore, you must ALSO achieve a sufficient score on the final exam, as shown in this table. Your final grade is be **limited** by both your total-points grade (above) and the grade in this table based on your final-exam percentage. Notice that these percentages are substantially more generous than the normal point scale.

Note that an 83 is not an A on the final. It is a B, but it still allows you to get an A for the course. Similarly a 55 is not a C-. It is an F. But with it

you can still earn a C- in the course. And even if you get a zero on the final, you can still earn a D in the course.

83+	A	80-82	A-	77-79	B+
73-76	B	70-72	B-	65-69	C+
60-64	C	40-59	C-	20-39	D+
0-19	D				

By way of explanation, I have had some students who did very well on the labs, sometimes incredibly well, but did very poorly on the exams, sometimes incredibly poorly. I discovered that one such student was just very good at getting other people to help him, but never did develop the skill to do the job himself. The labs are meant to be serious learning experiences. The exams are meant to verify that the learning took place. They are an “audit” of your knowledge. It would take too long to test everything you should know, so I try to pull out a representative sample of things you should know or be able to do.

3.1 In-Class Practices

12.5% of your grade is based on In-Class Practices. You need 125 points for a perfect score. There are actually 145 points available to work on.

I take roll in this class to help me learn your names. I also do some activity for points each class. Usually it is some sort of in-class practice where you demonstrate the ability to do something we have been talking about. These generally occur during the last half of the class hour. In the past I have given points for attendance, but I no longer do that because I am giving points for the in-class practices. On some days I may not have time for a practice. In that case I give full practice points to everyone that is in attendance. Missing and unnoticed persons get zeros, so be sure to alert me after class if you came in late.

3.2 Testing Center Quizzes

17.5% of your grade is based on Testing Center Quizzes. You need 175 points for a perfect score. There are actually 220 points available to work on.

There are several 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 sudden illness, discuss it with me.)

3.3 Programming Labs

40% of your grade is based on Programming Labs. You need 400 points for a perfect score. There are actually 450 points available to work on.

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. Maybe less. But you are not a professional. The work is difficult mostly 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.

Much like CS 101, 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.

Some of the programs will be graded by demonstration in class or for the tutor. In such cases you will have more freedom in the user interface. Programs graded by GradeBot allow freedom in the methods used to construct the answer, but they do not allow freedom in how the answer is presented: the user interface is specified for you.

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 web, “turnin,” or email.

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 “conversation” with GradeBot before you start writing your program.

If you discover a case where you believe that GradeBot 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.

Lab Submission Rules Cheating is sometimes a problem in this class. I have some rules. I am unhappy when I see cheating in any class. There have been 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. Your goal is to learn, not merely do things that might let you complete the assignments (without learning).

Open-Neighbor versus Copying: On all labs, I allow you to *confer* with other people, including other students. You may read their code (if they let you). You may show your code to them. You may talk about your code, your approach, your difficulties, and your ideas. You may draw pictures and make analogies and ask questions. You may use their ideas. However, *you must not make a copy of their code or submit their code to GradeBot, even if you first modify it.*

PLEASE: 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. Never give a disk or email a copy. 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**.

3.4 Midterms and Final

15% of your grade is based on three Midterm Exams (5% for each exam). You need 150 points for a perfect score. There are no extra points to work on. 15% of your grade is based on your final exam. You need 150 points for a perfect score. There are no extra points to work on.

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. Blank scratch paper is allowed. No books, no notes, and no calculators.

The purpose of the exams is to verify that you are learning the material that is involved with the labs. Because it takes a long time to write a program, I can only ask a few questions. Some material we learned may not show up on any of the tests. But if it does, you need to be ready for it.

3.5 Deadlines

Each assignment has a deadline. You can see these deadlines by asking GradeBot. Most lab 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.

3.6 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." A UW is worse than an F because it does not count as credits attempted.

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 time line for completion, and you get the necessary paperwork filled out.

4 Course Calendar

Lectures and in-class practices will match the assignments that are due. In-class tests will take one hour each (except the final). I have allowed an extra day (Tue, Thu, Sat instead of Mon, Wed, Fri) for many assignments, but you are encouraged to complete them earlier.

First block: During the first (approximately) ten hours of class, we will review everything we assume you learned in your previous programming classes. We will show you how to do these same things using Perl. There is a small program due almost every day. We end with the first midterm exam.

```
: hello          20  hello world
```

: celsius	30	convert Fahrenheit
: QP1	20	precedence simple
: QP2	20	precedence mixed
: phonecard	30	phone card comparison
: starbox	30	draw a box of stars
: lessthan	40	how many were less
: argv	30	command line access
: mid1	50	midterm 1 in class

Second block: During the second (approximately) ten hours of class, we focus on CGI and the web. There is a program due about every two days, but the programs are a bit harder. We also learn some regular expression processing. We end with the second midterm exam.

: QL1	20	CGI Language I: html, perl
: QR1	20	regular expressions
: QR2	20	regular expressions
: calc	40	tutor: calculator
: nuts	40	tutor: nuts, bolts
: mid2	50	midterm 2 in class

Third block: During the third (approximately) ten hours of class, we focus on DBI, the database interface, using SQL. We store, retrieve, and update information in databases. We create new tables. We build upon our CGI skills. We end with the third midterm exam.

: QS1	20	SQL I: commands
: QS2	20	SQL II: data types
: tv	30	Table Viewer
: QL2	20	CGI Lang II: incl dbi, sql
: buyone	40	Buy One
: mid3	50	midterm 3 in class

Fourth block: During the last (approximately) ten hours of class, we finish the major project for the semester, and we deal with programming complexity and simple software engineering. We study “big oh” analysis. This also gives us some time when labs are not due, so those that have gotten behind have a chance to submit late work before the end of the semester. We end with the final exam.

: manager	40	Manager
: shopcart	40	Shopping Cart
: QB1	10	big oh analysis (simple)
: checkout	30	Check Out
: QB2	10	big oh analysis (complex)
: final	150	final in class

5 Outcomes

An important way of looking at a course is to ask what outcomes it should create. Students completing this course will be able to do the following things:

- design, write, debug complex programs in a modern high-level programming language (e.g., Perl or Python).
- use control structures (if, while, sub).
- use abstract data types (array, stack, queue, hash).
- apply precedence to resolve complex math expressions.
- use html to create a web form with many input fields.
- use regular expressions to match, extract, and replace data.
- design, write, debug multi-path cgi programs in a modern high-level programming language (e.g., Perl or Python).
- write simple sql queries (show, create, select, insert, update, drop) and programs that use such queries.

6 Knowledge Units

The CS 201 course covers the following CC2001 Knowledge Units. These are defined in Computing Curricula 2001, a joint project of IEEE-CS and ACM. The IEEE Computer Society and the Association for Computing Machinery are the two major professional societies in computer science.

Much of this material is covered in more than one course.

PF1. Fundamental programming constructs

We review material that was introduced in CS 101.

- Basic syntax, semantics of a higher-level language
- Variables, types, expressions, and assignment
- Simple I/O
- Conditional and iterative control structures
- Functions and parameter passing
- Structured decomposition

PF2. Algorithms and problem-solving

- Problem-solving strategies
- The role of algorithms in problem-solving process
- Implementation strategies for algorithms
- Debugging strategies
- The concept and properties of algorithms

AL1. Basic algorithmic analysis

We introduce material covered more fully in CS 301.

- Asymptotic analysis of upper complexity bounds
- Big O and Θ (theta) notation
- Standard complexity classes (linear, log, exponential)

NC4. The web as an example of client-server computing

- Web technologies
 - Server-side programs
 - Common gateway interface (CGI) programs
 - Client-side scripts
- Characteristics of web servers
 - Handling permissions
 - File management
 - Capabilities of common server architectures
- Role of client computers
- Nature of the client-server relationship
- Web protocols
- Tools for web site creation and management
- Developing Internet information servers
- Publishing information and applications

IM5. Database query languages

We introduce material covered more fully in IS 350.

- Overview of database languages
- SQL (query formulation, update sublanguage)
- Embedding queries in a procedural language

7 Additional Statements

All syllabi are encouraged or required to address certain topics. These are generally considered to be common sense, but we find that it is useful to mention them explicitly anyway.

7.1 Accommodating Special Needs

I am personally committed to making this course as easy as possible (but no easier). To this end, I give many quizzes in the testing center without time limits. To fully teach important concepts, I give lab work, but it is not constrained by the amount of time available in class. I publish important assignments on my web site so you do not have to rely on note taking or memory to know what you need to do. I allow an unlimited number of attempts on labs that are graded by GradeBot, and make it available 24

hours a day throughout the semester. I believe that many cases of special needs are already accommodated by these practices.

For in-class examinations (three midterms and one final exam) I apply a strict time limit and do not allow outside resources. To avoid giving an unfair advantage, I require those who need a special accommodation to establish their rights by working through the BYUH Special Needs Coordinator.

Brigham Young University Hawaii is committed to providing a working and learning atmosphere which reasonably accommodates qualified persons with disabilities. If you have any disability that may impair your ability to complete this course successfully, you are invited to contact the Students With Special Needs Coordinator, Leilani A'una at 293-3518. Reasonable academic accommodations are made for all students who have qualified documented disabilities.

7.2 Plagiarism

<http://en.wikipedia.org/wiki/Plagiarism> has a wonderful article on plagiarism. Read it if you are not familiar with the term. Essentially, plagiarism is when you present the intellectual work of other people as though it were your own. This may happen by cut-and-paste from a website, or by group work on homework. In some cases, plagiarism may also create a violation of copyright law. If you borrow wording from someone else, identify the source.

Intentional plagiarism is a form of intellectual theft that violates widely recognized principles of academic integrity as well as the Honor Code. Such plagiarism may subject the student to appropriate disciplinary action administered through the university Honor Code Office, in addition to academic sanctions that may be applied by an instructor.

Inadvertent plagiarism, whereas not in violation of the Honor Code, is nevertheless a form of intellectual carelessness that is unacceptable in the academic community. Plagiarism of any kind is completely contrary to the established practices of higher education, where all members of the university are expected to acknowledge the original intellectual work of others that is included in one's own work.

In this course group work is permitted and encouraged but you are not allowed to turn in work that is beyond your understanding, whether you give proper attribution or not. Make sure you understand what you are submitting and why each line is there.

Faculty are responsible to establish and communicate to students their expectations of behavior with respect to academic honesty and student conduct in the course. Observations and reports of academic dishonesty shall be investigated by the instructor, who will determine and take appropriate action, and report to the Honor Code Office the final disposition of any incident of academic dishonesty by completing an Academic Dishonesty Student Violation Report. If the incident of academic dishonesty involves the violation of a public law, e.g., breaking and entering into an office or stealing an examination, the act should also be reported to University Police. If an affected student disagrees with the determination or action and is unable to resolve the matter to the mutual satisfaction of the student and the instructor, the student may have the matter reviewed through the university's grievance process.

7.3 Sexual Harassment

BYUH's policy against sexual harassment complies with federal Title IX of the Education Amendments of 1972 to protect university students from student-to-student sexual harassment both in and out of the classroom setting. Any incidents of such student-to-student harassment should be reported to either the Director of Human Resources (293-3713) or the Honor Code Office (293-3531). Allegations of sexual harassment are taken seriously. Upon receiving a report of sexual harassment, the Director of Human Resources will take appropriate action to resolve and correct conditions resulting from individual perceptions or from inappropriate behavior.

7.4 Syllabus is Subject to Change

It is possible that I will revise aspects of the course as we go along. 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.