COMP 171: Scripting Language Practicum (Section 001)
Fall 2009 Course Information & Syllabus

Instructor: R. I. Greenberg
Computer Science Department
Loyola University
820 N. Michigan Ave.
Chicago, Illinois 60611-2147

Phone: (773)508-3782 on Monday/Wednesday/Friday
Email: rig@cs.luc.edu **Home page:** http://www.cs.luc.edu/~rig
TA tutoring schedule: See `http://www.cs.luc.edu/academics/services/tutoring`.

Lectures: Tuesday 10:00–10:50 am in SKY-414.
Sometimes lecture notes or a summary may be available on the web. Other than that, if you have to miss a class, get notes from another student; mine are typically pieced together from more than one place with a lot of metacomments, which makes it hard for anybody but me to follow them. Also get copies of any missed handouts (available on the web site). The handouts are numbered sequentially, starting with handout 0. On handout 0, you need to fill in some information and return it to me promptly so you can be on the email list and get access to the web site for the course.

Office Hours: In DH-225: 12:30–1:30 pm on Monday, Wednesday, and Friday.
These are the guaranteed times to find me except as announced in advance. You should also feel free to look for me at other times or make appointments.

Course Objectives: The purpose of this course is twofold: (i) to provide a painless introduction to programming computers, and (ii) to serve as a basis for programming in applied disciplines such as biology, chemistry and business. Traditionally, programming languages can be categorized as compiled or interpreted as also typed or typeless (actually more- or less- typed). Scripting languages usually fall into the category of interpreted and less-typed. Scripting languages owe their speed of development to close tie-ins with native operating system libraries. Popular scripting languages include all of the Unix shell scripts (csh, tcsh, zsh etc.), Tcl/Tk, MS-DOS command language, Perl, Python, etc. Lately Perl and Python have become popular due to their simplicity, expressiveness and applicability to various scientific and business domains. After taking this course, students should be able to: 1. Understand what scripting languages can do and when they are suitable for use. 2. Program in Perl. 3. Program in Python.

Prerequisites: None (but facility with basic high school mathematics is expected).

Textbook: Harvey M. Deitel, Paul J. Dietel, Tem Nieto, and D. C. McPhie. *Perl How to Program*. Prentice-Hall, 2001. (Cover also notes "Introducing CGI and Python".)

Course Requirements: There will be several homework assignments, a midterm, and a final. The weightings within the semester grade will be: Homework 45%, Midterm Exam 20%, and Final exam 35%.

Homework: Only homework turned in by the due date is guaranteed to be graded. Any special circumstances that cause difficulty in meeting the deadlines should be brought to the attention of the instructor in advance. Homework must be handed in at the beginning of class, since solutions may be discussed in the same class on occasion. Homework turned in to my mailbox will generally not be graded, since I do not check the box continually and cannot generally verify that homework was turned in before solutions were discussed in class. If you cannot turn in homework in person, you should put it under the door of my office.

Exams: The midterm exam, tentatively scheduled for **??** is 50 minutes long. The final exam is scheduled for 1:00–3:00 pm on Tuesday, December 8.

Collaboration: *No* collaboration is permitted on exams. *Collaboration* on homework is acceptable, but *copying* is not! (Safeguard your files and printouts.) You may discuss solution techniques with other students, but you must write up your solutions independently. If you obtain a solution through research, e.g., in the library, credit your source and write up the solution in your own words.

Tentative Course Outline and Approximate Schedule:

Recommended readings from the text are shown on a weekly basis. (When selected sections or subsections are listed, it is assumed that you will include the introduction of the corresponding chapter or section.)

1. (8/25) Introduction to Perl Programming. Sections 2.1–8
2. (9/1) Introduction continued. Sections 2.9–12. Control Structures Part 1. Chapter 3.
3. (9/8) Arrays. Sections 4.1–10. Hashes. Section 4.13–14.
4. (9/15) Control Structures Part II. Sections 5.1–13.
5. (9/22) Catch-up or Review. Subroutines and Functions. Sections 6.1–6.
6. (9/29) Exam I on Chapters 1–5.
7. (10/13) Recursion. Section 6.10. Miscellaneous topics. Sections 6.12–15. Introduction to Common Gateway Interface (CGI). Sections 7.1–5 and 7.7.
8. (10/20) Regular Expressions. Sections 8.1–11.
9. (10/27) String Manipulation. Sections 9.1–10. 9.11–12 time permitting.
10. (11/3) File Processing. Sections 10.1–6. File and Directory Manipulation. Sections 11.1–4.
11. (11/10) Python. Sections 27.1–5 and 27.12.
12. (11/17) Python catch-up and BioPython.
13. (11/24) BioPerl.
14. (12/1) Review.