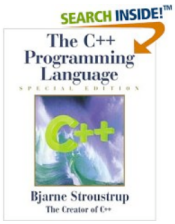
Syllabus for
Programming Languages
Computer Science
COMP 335

Instructor: Mark LeBlanc (mleblanc)

Office: SC-B103
Phone: 286-3970 (on campus: x3970)

Office Hours: by appt. *or*
Mon 9:30-10:30, 3:30-5:00
Wed 9:30-10:30, 3:30-5:00
Fri 9:30-10:30
Meeting: MW 2:00-3:20pm
A118 or csLab

Required Texts:
(1) *An Introduction to Object-Oriented Programming (3rd edition).* Timothy Budd, Addison Wesley, 2002.



Additional Readings:
(2) Selected readings from *Dr. Dobbs Journal*.

(3) *Concepts of Programming Languages* (Ch. 3) by Robert Sebesta, Benjamin Cummins, 2001.

(4) Perl for Exploring Biological Sequences by LeBlanc and Dyer, Oxford University Press, (late 2006).

You will continually need to augment this text with language references from the library. Our library (as well as csLab and my own library) is well stocked with Perl, Visual Basic, Lisp, C, C++, Fortran, Cobol, Prolog, Modula, Java, Pascal, Scheme, Ada, assembler, and other titles.

Good web sites:

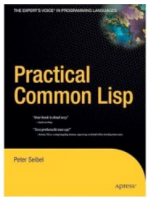
Dr. Dobb's Journal:
http://www.ddj.com

99 bottles of beer -- One program in 880 languages
http://99-bottles-of-beer.ls-la.net/

Computer Languages History
http://www.levenez.com/lang/

The Language List
http://people.ku.edu/~nkinners/LangList/Extras/langlist.htm

An Open Directory Project index:
http://dmoz.org/Computers/Programming/Languages/



How to write a parser: LEX and YACC
http://members.tripod.com/~ashimg/Parser.html

Good C/C++ Reference
http://www.cppreference.com

Content:

This course provides an introductory theoretical study of the principles that govern the design and implementation of contemporary programming languages. This includes language syntax (lexical properties, BNF notation, and parsing); language representations (data structures, control structures, and runtime environments); and language styles (procedural, functional, declarative/logical, scripting, and object-oriented). Particular focus will be on system and object-oriented languages. Labs, homeworks, and programming assignments will include the following languages:

Language type	**we will do**
system tools	C, bash
object-oriented:	Java and C++
functional:	Lisp
declarative (logic):	Prolog
data-interface/scripting:	Perl

Your grade:	5 Homeworks	**15**%	continual throughout the semester
	6 Programs	**45**%	continual throughout the semester
	3 Take Home Exams (10% each exam)	**30**%	Feb 13 (due Fri, Feb 17th), Feb 27 (due Wed, Mar 8th) Mar 27 (due Fri, Apr 7th)
	Team-Led Labs	**10**%	TBA (last three weeks of classes)
	Day #1 "hello world"	5%	
	Day #2 walk thru/demo	5%	

Honor Code Revisited: **It goes without saying that all submitted work will be the student's own, in keeping with the Wheaton Honor Code.** For labs, you may get help from fellow classmates, but remember that **all submitted work must be your own.** For homework, **the programs and hand-written work must be your own from beginning to end**.

Week	Monday	Wednesday
1		Jan 25 **History I**
2	Jan 30 **History II** **C and UNIX (Part I)** *Homework #1 (languages) due in class*	Feb 01 **C and UNIX (Part II)** *Homework #2 (Perl progress) due in class*
3	Feb 06 **Describing Syntax and Semantics** formal languages and grammars (Sebesta handouts) *a1 (`tail` utility) due* Math/CS "Chats" -- 8pm, A102, pizza++ *Homework #3: just attend and ask a question*	Feb 08 **Compilers** BNF form, parse trees, syntax graphs *Homework #4 (Team Lab Agenda) due in class* *a2 (`bitFu`) due on Friday, Feb 10*
4	Feb 13 **EBNF** ambiguous grammars, recursive descent parsing *Take Home Exam I (due Fri, Feb 17)*	Feb 15 **Designing a grammar** Together we design a language *Take Home Exam I due Fri, Feb 17*
5	Feb 20 **Recursive Decent Parser (Part I)** *a3 (spider) due*	Feb 22 **Recursive Decent Parser (Part II)**
6	Feb 27 **Object-Oriented Design** (Budd text) *Take Home Exam II (due Wed, Mar 8)*	Mar 01 **no class** *Mark is presenting at SIGCSE 2006* *a4 (parser) due*
7	Mar 06 **Object Syntax** classes, methods, messages, case studies (Budd text)	Mar 08 **Inheritance/ Software Reuse** (Budd text) *Take Home Exam II due*
8	Mar 13 **Spring Break**	Mar 15 **Spring Break**
9	Mar 20 **Polymorphism I**	Mar 22 **Polymorphism II** *a5 (Java) due*
10	Mar 27 **Bash** *Take Home Exam III (due Fri, Apr 07)*	Mar 29 **Bioinformatics and Perl** (2-5pm) *"Linked" lab with BIO 316 students – Hmwk #5* *a6 (Zoo in C++) due*
11	Apr 03 **Functional Programming in Lisp** *Homework #5 (Bioinformatics) due in class*	Apr 05 **More Lisp** *Take Home Exam III due Fri, Apr 07*
12	Apr 10 **Logic Programming in Prolog**	Apr 12 **More Prolog**
13	Apr 17 **Team I lab**	Apr 19 **Team I lab**
14	Apr 24 **Team II lab**	Apr 26 **Team II lab**
15	May 01 **Team III lab**	May 03 **Team III lab**

PHILOSOPHY of COURSE

This course will run very much like a graduate school course. That is, the pace will be quick, the workload heavy, a strong dose of introductory theory applied throughout, and an **air of professionalism** maintained by both the instructor and the student. Said differently, the instructor places high expectations on your participation in class and equally high expectations on the amount of *independent* effort you apply. NOTE: *The exact pages to be assigned for reading will be given in lecture.*

WRITTEN or PROGRAMMING HOMEWORK

It is expected that you spend *at least* **2 hours** on reading and practice problems for every hour of lecture. This computes to *at least* **6 hours of work in the texts per week**. This course is *not* just about learning new languages. Rather the focus is on the theoretical basis for programming languages and on systems and object-oriented languages in particular. Let there be no misunderstandings: you *must* spend time in the texts and associated articles -- reading, rereading. There are approximately 5 homework assignments. Programs must be electronically submitted by 5am of the day after the due date (yes, you get 5 extra hours if you need and are willing to stay up ☺). A program beyond 5am and one day late will be docked 10%. No programs can be submitted after more than one day late. Of course, one day late extends for 24 hours after the 5am deadline. Assume this deadline is fixed. There are six programs in total.

TAKE HOME EXAMS

There are three take home exams during the semester. You may *not* work together on take home exams.

PERL BOOK

Throughout the semester you will be independently learning Perl using the manuscript entitled: "Perl for Exploring Biological Sequences" by LeBlanc and Dyer (Oxford University Press, to be published late 2006). The focus of the book is to teach novice programmers to learn to program in Perl. Note that the audience for the book is someone who has never programmed, so although I am expecting that you will complete the entire book, you will be able to work through the material very quickly. I will include milestones to push you along as part of some of the homework assignments and/or take home exams.

"Linking" with Biology

During the semester, we will have one lab with biology students. The students will be from Shawn McCafferty's course: BIO 316 – "Molecular Biology and Biotechnology". This lab (Wed, March 29) will run from 2 to 5pm. (We will cancel lecture on another day to account for the extra time expected from you; if you have a conflict with this extra lab, please speak to me immediately). During the lab, you will work on the web with a biology student on a bioinformatics problem. Your part of the endeavor will be to assess the type of data obtained during the lab and to *suggest a means of using Perl to parse, collate, sort, and generally morph the data into a more usable form for your biology partner.* For example, your biology partner might generate tens of files of data, but she only wants an analysis on one part of each file. As you'll see, Perl is a great tool for this type of task. One of your homework assignments will be to document after the lab what you can do for your partner. One section of your third take home exam will be a Perl program that delivers an analysis and a report that documents your contribution to your biology partner. Since these students will be engaged in their own semester-long research projects, you have the potential to participate in research through this part of the course.

Team-led LABS

You will join one or two other students (groups of two or three per group) to lead two 100-minute labs on a unique language that has not already been covered in class (e.g., R, PHP, Visual Basic, Scheme, etc). Milestones for selecting your language and preparing for these labs will be requested throughout the semester in some of your homework assignments. In the first lab, you will lead the other students and the instructor through a small set of lectures at the whiteboard each followed by a "hello world" hands-on programming example. This implies that you must have at least enough computers installed with the necessary language and programming environment. The second day/lab you may continue with one or two additional small lessons and then you will demonstrate a semi-large project that you jointly developed over the semester. Some of the time in this second day will be spent in a walkthrough of some of the software written. I will demonstrate the format by example during our weeks with Lisp and Prolog. For each of the two labs, I will allot your group a certain number of points and then the group members will decide on how to share the points. For example, a very strong lab between a pair of students might be assigned 180 points. Between them, they might decide one person took the lead so they agree to split the points 94 and 86. You are expected to deliver a polished set of two, professional labs.