CS010 – Introduction to Computer Science I
Spring, 2007

Course Information

OVERVIEW: The course introduces students to the field of Computer Science via *programming*. We will focus on the language *Scheme* without assuming any prior programming experience. Using Scheme programming as our vehicle, we will tour various concepts from CS including: design, logical thinking and problem solving, algorithm development, abstraction, software development process, and the object-oriented paradigm. Although the majority of time will be spent learning and using Scheme, the focus throughout will be on learning foundational concepts of problem-solving and design.

> "No prior computer or programming experience required. Basics of programming including language features, disciplined programming style, and documentation. Problem solving, algorithm design, and the software development process." (from the Westmont catalog)

PREREQUISITES: Fulfillment of the mathematics competency requirement.

COURSE OBJECTIVES:

- Learn disciplined design skills
- Be able to reason and problem-solve using abstractions (GE, II.D.)
- Implement solutions to problems as a Scheme program

In order to solve a problem of any significance, you must first understand the problem at a fairly deep level and then you must *design* a series of actions that embody a solution to the problem. In this class, you will learn and exercise design skills. As one of our tools, we will use *abstraction* to make complex problems manageable. This class satisfies the *reasoning abstractly* requirement of the GE. In the context of this class, abstraction takes several forms including functions, meta-functions, modeling, information hiding, hierarchical decomposition, and replication and repetition.

This class also provides an entry point to the major and minor in Computer Science. As such, students should: begin to learn to *think creatively and flexibly*; improve *communicating effectively*; start acquiring the *fundamental concepts and skills* of Computer Science; and identify initial *relationships between Christian faith and Computer Science* (departmental learning stardards). In addition, students should learn to identify the promise and peril of technology and discern appropriate technologies for particular problems (Westmont Learning Standard 6).

The successful student in this course will have the programming skills to take moderately scaled problems, formulate an algorithmic solution, and implement that solution in Scheme. Such students will also have acquired the conceptual framework for continuing study in Computer Science.

LOCATION: Voskuyl Library 106

DAYS & TIME: Tuesday and Thursday, 1:15-3:05 pm.

TEXT: *How To Design Programs: An Introduction to Programming and Computing*, by Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, & Shriram Krishnamurthi. MIT Press: Cambridge, Mass. (2001).

COURSE WEBPAGE: http://www.westmont.edu/~iba/teaching/CS010

Instructor Information

INSTRUCTOR: Dr. Wayne Iba
OFFICE: Mathematics and Computer Science Modular Building
OFFICE HOURS: TBD
EMAIL: iba@westmont.edu
PHONE: (805)565-6799

Course Assignments, Requirements and Policies

EXPECTATIONS: This class requires a greater than average time commitment for many students. If you are unable or unwilling to invest that time, you will probably struggle; please reevaluate your priorities at the beginning of the semester. Before you come to class, I expect you to be familiar with the sections of the text to be covered each day and to be prepared to contribute to the problems we solve in class. You need to bring your laptop to class each day (and power adapter if needed). Our department has some laptops that you may borrow for the semester if you do not have a laptop. Perhaps most importantly, I expect you to ask questions if you do not understand something. Asking questions is the key to the intellectual life. Besides, there is no future in remaining confused.

EXERCISES: Exercises are an ideal way to learn the skills and concepts that are the focus of this course. You are expected to have the competence to solve every exercise at the end of each section of your text, although a subset of problems will be assigned. Plan to turn your solutions in well before the deadline as late work may not be accepted. (Exceptions may be made at the sole discretion of the instructor.) Using the DrScheme environment, you will know for yourself if your solutions work or not. I will review a subset of the exercises in class.

PROJECT: During the latter portion of the semester, students will be given a more substantial programming assignment that will involve several phases of development. Each phase will have an associated deliverable that will be submitted for grading; together, the project deliverables will represent the primary component of the overall grade for the course. Failure to complete one or more installments of the project will significantly impact your grade in a negative manner.

TESTS & EXAMS: There will be five to ten exams during the semester and, a comprehensive final exam. Your lowest exam score will be ignored. Exams must be taken at the scheduled time unless arrangement is made with the instructor *in advance*.

ATTENDANCE: To learn a language, you must use it. In this respect, Scheme is no different from French. In the style of language immersion, we will go over numerous examples and you will write programs during class time. Therefore, attendance is required. If you have a reason for missing more classes than allowed by Westmont policies, discuss the matter with me *in advance*.

GRADING: Students will be evaluated on how well they master the skills of program design as demonstrated using the Scheme programming language. Both correctness and style are important and will be evaluated on exams and the project. A final weighted score will be based on the following: project (40%), collective exams (30%), exercises (30%). The final weighted score, will yield a letter grade according to the standard scheme where 0.9 and above yields an A, 0.8 to 0.9 a B, and so on. I reserve the right to lower the thresholds for some or all of the letter grades.

ACADEMIC HONESTY: As in every area of life, I expect that you will conduct yourself honestly within the context of this class. You are expected to have read and agreed to the Academic Dishonesty policy as part of the general Westmont Academic Policies, as well as the specifics of the policy on Plagiarism. Do not attempt to receive credit for work that is not your own without properly acknowledging sources via appropriate citations or references. You are encouraged to get help from your peers, but you must acknowledge such help (both received and given) and that you understand the issue on which help was received. Except where explicitly provided or allowed by the instructor, you must not use electronic copies of any portion of code for the exercises or project (unless distributed by me). The consequences of violating the trust I implicitly extend to you will be according to the Westmont policy; but more serious will be the damage done to our academic relationship.