CSCI 335 SYLLABUS

COURSE NAME, NUMBER	CSCI 335: Software Development
SEMESTER, YEAR	Spring 2011
INSTRUCTOR	T. VanDrunen
OFFICE / TELEPHONE / EMAIL	Sci 163 752-5692 Thomas.VanDrunen@wheaton.edu
OFFICE HOURS	MWF 1:30–3:30 pm; Th 10:20–11:30 am
COURSE WEBSITE	`http://csnew.wheaton.edu/~tvandrun/cs335`

RESOURCES

Jalote, *A Concise Introduction to Software Engineering*, Springer, 2008.
Gamma et al, *Design Patterns*, Addison-Wesley, 1994.
Brooks, *The Mythical Man-Month*, Anniversary Edition, Addison-Wesley, 1995.

COURSE DESCRIPTION

Principles and practices of software development including design patterns, validation and testing, and coordination of team projects. Introduction to data bases and user interface design. Professional issues in computing.

INFORMAL DESCRIPTION

This course has two purposes: it covers the principles of software development (sometimes called *software engineering*), hence the name of the course; it also is a landing place for various topics that need to be covered in the CS core curriculum, but don't fit nicely in any other course. From the first purpose we have concepts and terms, UML, group dynamics, project and team management, methodology, software life cycle, etc); design patterns; professional and ethical issues; readings, including *The Mythical Man-Month* and "The Cathedral and the Bazaar"; and some discussions about career path options. From the second purpose we have the basics of database use and design; user interface design, plus GUIs and event-driven programming; regular expressions, automata, and formal languages; and, time permitting, programming languages, distributed computing, web programming, scripting languages, and the history of the computing industry.

GOALS AND OBJECTIVES

1. Students will gain experience developing large software projects, including working on a software team.
2. Students will be able to understand and use the standard terminology and documentation tools for object-oriented software development.
3. Students will be able to articulate the fundamentals of database design and management, languages and automata, and language systems.
4. Students will be able to employ design patterns in their design of software systems.
5. Students will give thought to their career path and plan accordingly.

Grading:

	weight
Project checkpoint 1	15%
Project checkpoint 2	20%
Project final submission	25%
Peer evaluation of project contribution	5%
Short exercises and participation	7.5%
Presentation	7.5%
Midterm	10%
Final	10%

SPECIAL EXPECTATIONS

Academic Integrity

Since all programming in this course will be group work, academic integrity concerns are not about code sharing among students. Instead, students must be careful to document and cite all ideas that come from outside sources. Failure to do so may result in point deduction or rejection of the project altogether.

Late assignments

If a programming project is running late, the group should negotiate with the instructor before the due date (ie, the negotiation should take place before the day on which it is due) to determine a revised due date and a point deduction. Short exercises will not be accepted late.

Attendance

Students are expected to attend all class periods *on time.* It is courtesy to inform the instructor when a class must be missed.

Examinations

The final exam is Tuesday, May 3, at 8:00 AM. I do not allow students to take finals early (which is also the college's policy), so make appropriate travel arrangements. See the course website for the date of the midterm.

Special needs

Whenever possible, classroom activities and testing procedures will be adjusted to respond to requests for accommodation by students with disabilities who have documented their situation with the registrar and who have arranged to have the documentation forwarded to the course instructor. Computer Science students who need special adjustments made to computer hardware or software in order to facilitate their participation must also document their needs with the registrar in advance before any accommodation will be attempted.

Dress and deportment.

Please dress in a way that shows you take class seriously—more like a job than a slumber party. (If you need to wear athletic clothes, for example, because of activities immediately before or after class, that's ok, but try to make yourself as professional-looking as possible.) If you must eat during class (for schedule or health reasons), please let the instructor know ahead of time; we will talk about how to minimize the distraction.

Electronic devices.

Please talk to me before using a laptop or other electronic device for note-taking. I will discourage you from doing so; if you can convince me that it truly aides your comprehension, then I will give you a stern warning against doing anything else besides note-taking. Trying out programming concepts on your own during classtime is not productive because it takes you away from class discussion; moreover, that is what lab time is for. You cannot multi-task as well as you think you can. Moreover, please make sure other electronic devices are silenced and put away. ***Text in class and DIE.***

Office hours. I try to keep a balance: Stop by anytime, but prefer my scheduled office hours. This semester I am trying to reserve Tuesdays for uninterrupted work. This means that if it is possible to hold your question for another day, then that would be great, but if it is urgent (ie, holding up your progress on a project), then no problem, stop by even if it's a Tuesday. Also, any time my door is closed, it means I'm doing something uninterruptable, such as making an important phone call. Do not knock; please come back in a few minutes or send me an email.

Projects Most of the work in this course will be on a group project in three phases. In the first two phases of the project, you will work in teams of two or three, chosen randomly. The teams will be different for each phase. In the third phase, you will all work together as a single team. Most of the grade for the project will be given in common to all members of the team; however, part of the evaluation of the final phase will involve teammates evaluating each other's contribution to the project. In total, the project will count for 65% of a student's grade in this course. Dates for completing the checkpoints are, approximately, Feb 25, Mar 25, and Apr 27.

Textbooks and readings. This semester I have adopted a new textbook for giving an overview of software development, Jalote's *A Concise Introduction to Software Engineering.* We will use this mainly in the first half (especially the first 3 weeks) of the course. Students should plan to read the indicated chapter before class; I will assign a short set of questions or a summary to enforce this. We will discuss the material as a class.

The two other textbooks for this course are important texts which the students should plan to keep. Gamma et al is the standard source for the material covered in the design patterns theme of the course. Students should decide for themselves how they can use the text most effectively: some may choose to read the relevant sections prior to their presentation in class, some may choose to read them after, some may choose to skim first and read carefully after. But by all means students are responsible for the material in them, and mastery of the material will not be obtained without the combination of class presentation, textbook reading, and participation in related exercises.

Brooks' *The Mythical Man-Month*, will be the last topic covered in the course. Selections from this book (which will be supplemented by excerpts from Brooks' more recent book, *The Design of Design*) must be read prior to class discussion on the material. In past semesters I have found it to be too much of a burden on students to read so much material while also finishing their project, so I will assign these readings earlier in the semester, even though we will discuss them at the end.

Library. In the bookcase of the computer science lab, there is a dedicated shelf of books providing additional resources for this course. You are referred to those books for fuller treatments of some of the topics touched on in this course.

Short exercises. On most class days you will be given a short exercise or a response to a reading to reinforce the concept presented or discussed in that class period or prepare you for the next. Short exercises should be turned in by email (unless otherwise noted) before the start of the next class period. These exercises are primarily instructive, not evaluative. Your grades on these are mainly credit for doing the assignment, though a lack of effort or thoroughly incorrect answers will be penalized. Moderate collaboration on these exercises is permitted, though in exercises that require writing code, each student must turn in code written independently.

Presentation. Each student is required to make a presentation on one topic during the course of the semester. The student will present for about one half of the course period (or about 30 minutes). Each student must meet with the instructor twice prior to the presentation: A first meeting for the instructor to advise the student on resources, a second meeting for the student to show a detailed outline to the instructor and describe any slides, handouts, or related material he or she intends to use. Each student must choose a presentation topic no later than Friday, January 21. Topics available for presentation are the Python programming language (Feb 9), XML (Feb 28), Apps for mobile devices (Mar 23), and the PHP programming language (Mar 30). Each topic may be chosen by up to two students, who will coordinate coverage of topics and *each* make a 30-minute presentation (or tag-team for a complete hour).

Content. See the course website for the sequence and schedule of topics.