CSCI 290 – Fall 2008 Handout: Syllabus
Object-Oriented Software Design
Michael H. Goldwasser
Saint Louis University Tuesday, 26 August 2008

CSCI 290: Object-Oriented Software Design

Contents

1 Overview

1.1 Catalog Description

> ”An implementation-based study of object-oriented software development. Teams will design and create medium-scale applications. Additional focus on the design and use of large object-oriented libraries, as well as social and professional issues.”

1.2 Prerequisites

The official prerequisite is CSCI 180.

1.3 Detailed Description

The primary focus of the course is the process of *designing* software using object-oriented principles. The goal is to provide knowledge and experience when decomposing a project into a cohesive set of software components. With good design, those components can readily be implemented, maintained, and reused. With experience, successful design patterns can be recognized and applied.

In our programming sequence at SLU, this course is somewhat of a bridge between the introductory courses CSCI 150 and CSCI 180, and more advanced courses such as Software Engineering. The earlier courses introduced object-oriented principles and mechanisms, but there was relatively little emphasis on the top-level design process. Small projects were typically completed with only a few necessary classes. With larger project the instructor typically prescribed the set of classes and interfaces while the student was responsible for the low-level implementation. In this course, students will focus more on design stages necessary for medium- or large-scale project.

Most aspects of object-oriented design are independent of a particular programming language. That said, we will want to see the high-level designs progress to a low-level implementation. We will rely upon the students' presumed familiarity with both Python and C++ from the prerequisite course sequence. Yet we are also choosing to introduce students to programming in Java, and will make an effort to highlight the similarities and differences between Python, C++, and Java in their support of object-orientation. Our exploration of Java and its remarkably large API will have a secondary benefit of providing concrete examples of large object-oriented hierarchies and common design patterns.

2 Course Administration

2.1 The Staff

Instructor:	Dr. Michael Goldwasser
Email:	`goldwamh@slu.edu`
Web:	`http://cs.slu.edu/~goldwasser/`
Office:	Ritter Hall 006
Telephone:	(314) 977-7039
Office hours:	(generally in RH 006, but occasionally in RH 121) Mondays 10:50–11:50am Wednesdays 2:00–3:00pm Thursdays 3:30–4:30pm or by appointment

Please make sure to take advantage of office hours, as they offer a wonderful opportunity for individual interaction.

2.2 Class Meetings

The Lectures

Class meetings will be far more interactive than those in the introductory courses. Much of our time together will be devoted to group discussions and design exercises. *For this reason, attendance at every class is essential.*

Time:	Tues/Thur, 2:15–3:30pm
Place:	Ritter Hall 121

2.3 Textbook

The required textbook for this course is:

Title:	*Objects-Oriented Design & Patterns, 2nd edition*
Author:	Cay S. Horstmann
Publisher:	John Wiley & Sons, 2006
ISBN-13:	978-0-471-74487-0
ISBN-10:	0-471-74487-5
Website:	`www.horstmann.com/design_and_patterns.html`

The text should be available through the campus bookstore as well as various online book vendors.

3 Online Resources

This course will take advantage of the Internet and the departmental network in many ways.

3.1 CSCI 290 Web Page: `cs.slu.edu/~goldwasser/290/`

With the exception of the first day's printed handouts, most of the information for this course will be distributed only by means of the course web page. This web site will contain all assignments, a schedule of topics, and links to other sources of documentation.

The web page contains some information (e.g. solutions, submitted assignments, individual grades) that is more sensitive and therefore which will be available to students in the class only after they have verified their identity. To gain access to these parts of the web page, a student must first complete an online questionnaire, creating a unique identity and password.

3.2 Electronic Assignment Submission

All assignments for this course must be submitted electronically! The submission procedure will be done through the course web page, and allows students to submit from any computer connected to the Internet. Each student in this class will be selecting a unique username/password combination solely for use in identifying the student when using the course web page.

3.3 Email with Instructor

Face-to-face contact in class and in office hours is most desirable. Yet email is a convenient form of communication as well. I try to respond to email promptly, including at least once each evening when possible.

If your question involves your progress on a current programming assignment, my response will be more informative if you can point out the specific problem you have encountered, and if I am able to see all of your source code. Therefore I strongly suggest that you either attach all relevant files to the email or submit preliminary versions of such files through our online system.

4 Graded Work

4.1 Assignments (40%)

There will be a series of assignments covering various aspects of design and implementation. Many of these will involve group work. Details will be provide as the course progresses.

4.2 Exams (60%)

- **Midterm Exam (20%)**, Thursday, 16 October 2008, 2:15–3:30 p.m
- **Final Exam (40%)**, Thursday, 16 December 2008, 2:00–3:50 p.m

4.3 Course Grades

Letter grades will be based on each student's overall percentage of awarded points according to the following formula.

Student percentage above 90% will result in a grade of A or better.
Student percentage above 87% will result in a grade of A- or better.
Student percentage above 83% will result in a grade of B+ or better.
Student percentage above 80% will result in a grade of B or better.
Student percentage above 77% will result in a grade of B- or better.

Student percentage above 73% will result in a grade of C+ or better.
Student percentage above 70% will result in a grade of C or better.
Student percentage above 67% will result in a grade of C- or better.
Student percentage above 60% will result in a grade of D or better.
Student percentage below 60% will result in a grade of F.

Any modification to this scale at the end of the year will be *in favor of the students.* That is we may later decide to award an A to a student who is slightly below the above cutoff, but we certainly will not deny an A from someone who is above the cutoff.

4.4 Academic Integrity

Students are expected to have read and abide by the University statement on Academic Integrity available on page 58 of the Saint Louis University's Undergraduate Catalog. A more detailed policy statement is given by the College of Arts & Science (`www.slu.edu/colleges/AS/academic_honesty.html`), also applying to this course.

In addition to those general statements, we wish to discuss our policy in the context of this course. When it comes to learning and understanding the **general course material**, you may certainly use other reference materials and you may have discussions with other students in this class or other people from outside of this class. This openness pertains to material from the text, practice problems, general syntax and use of a programming language or other computing tools.

However, when it comes to **work that is submitted for this course**, you are not to use or to search for any direct or indirect assistance from unauthorized sources, including but not limited to:

- past students, whether from this school or other schools
- other acquaintances
- other texts or books
- online information other than that referenced by course materials

Acceptable sources of information include consultations with the instructor, as well as any materials explicitly authorized in an assignment. Even in these cases, if you receive significant help you should make sure to document both the source of the help as well as the extent.

For most of the assignments in this course, students will be encouraged to form groups. In this case, conversations between partners is both permissible and required, but conversations with members of other groups is still forbidden. Furthermore, all students within a group are expected to contribute significantly to the development of the submitted work. It is unethical to allow a partner to "sign on" to a submission if that partner did not significantly contribute to the work.

Any violations of these policies will be dealt with seriously. Penalties will apply as well to a student who is aiding another student. Any such violations will result in a minimum penalty of a zero on the given assignment that cannot be dropped, and severe or repeated violations will result in an immediate failing grade in the course. Furthermore all incidents will be reported in writing to the Department and/or the Dean, as per the College procedure.

4.5 Late Policies

All exams must be taken promptly at the required time. Requests for rescheduling an exam will only be considered if the request is made prior to the start of the exam, or else in an "emergency" situation with appropriate documentation.

For assignments, we wish to allow students to continue to work comfortably beyond the official deadline when a little more time will result in more progress, while at the same time discourage students from falling significantly behind pace and jeopardizing their success on future assignments. Our solution is the following exponentially decaying late formula (some have suggested that we should offer extra credit to anyone who fully understands this formula).

We will consider an assignment submission "complete" when any part of the assignment is last submitted or modified. Any assignment that is not complete promptly by its due date and time will be assessed a penalty based on the formula $S = R \cdot e^{-h/173}$, where S is the grade given, R is the grade the work would have received had it been turned in on time, and h is the amount of time (in hours or fractions thereof) that the work was late. Examples:

— work turned in 1 hour late receives over 99.6% of its original credit
— work turned in 5 hours late receives over 97% credit
— work turned in one full day late receives less than 88%
— work turned in two full days late receives less than 76%
— work turned in five days late receives less than 50%

The above policies will be waived only in an "emergency" situation with appropriate documentation.

5 Additional Information

5.1 Students with Disabilities or Special Needs

In accordance with the Americans with Disabilities Act, reasonable accommodations may be made to assist a student with a documented disability.

Any student who feels that he/she may need academic accommodations in order to meet the requirements of this course, as outlined in the syllabus, due to presence of a disability should contact the Office of Diversity and Affirmative Action. Please telephone the office at 314-977-8885, or visit DuBourg Hall Room 36. Confidentiality will be observed in all inquiries.

5.2 Computing Resources at SLU

Our department runs a computer server named `turing` that serves as the primary computing environment for this course. If you do not already have an account on this machine, please let us know and we will create a new one.

See `cs.slu.edu/computing-resources` for further documentation regarding use of turing's facilities.

Please note: you are not explicitly required to use `turing` as your computing platform, it is simply the only platform what we will officially support. If you wish to use another platform that offers you sufficient support for completing your assignments, please feel free to do so.

5.3 Related Reading

For the general issue of design patterns, the classic resource is the so-called ”gang of four” book,

Title:	*Design Patterns: Elements of Reusable Object-Oriented Software*
Author:	Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides
Publisher:	Addison Wesley, 1995
ISBN-13:	978-0-201-63361-0
ISBN-10:	0-201-63361-2

You might also be interested in acquiring a language reference book for Java, although some prefer to rely on the wealth of online materials. If you do purchase a book, please make sure that it covers Java version 5 or later (the current version of Java is 6, however most significant were changes between 4 and 5). Some recommended titles include:

Title:	*Java: The Complete Reference (7th edition)*
Author:	Herbert Schildt
Publisher:	McGraw Hill Professional, 2006
ISBN-13:	978-0-07-226385-5
ISBN-10:	0-07-226385-7

Title:	*Java How to Program (7th edition)*
Author:	Harvey Deitel and Paul Deitel
Publisher:	Prentice Hall, 2007
ISBN-13:	978-0132222204
ISBN-10:	0132222205

Title:	*Java in a Nutshell (5th edition)*
Author:	David Flanagan
Publisher:	O'Reilly Media, 2005
ISBN-13:	978-0-596-00773-7
ISBN-10:	0-596-00773-6