Location

MWF 1205-1255

Klaus 1443

Class Objective

Purpose:

CS2340 takes students who know an object-oriented language, and focuses on getting them to use that language in a true object-oriented style. The course achieves this goal by introducing a design methodology and notation, and covering standard principles and practice in design.

Outcomes:

(Movement - Synthesis) Improve existing programming skills by developing much larger and more complex programs than in previous classes.

(Accomplishment - Synthesis) Given a requirements list, complete a four person large-scale programming project that implements those requirements. The project will require at least 5000 lines of code and multiple compilation modules (or equivalent jars) to complete.

(Experience - Analysis) Reflect on the difficulties of team membership and the challenges of developing software in a team environment.

(Competency - Application) Demonstrate the ability to use a version control system such as Subversion to manage team code.

(Competency - Application) Demonstrate the ability to use standard tools to help with large-scale projects such as automated build scripts (such as Ant), automated code checking (PMD, Checkstyle, FindBugs) and commercial quality development environments (such as Eclipse).

(Movement - Synthesis) Improve object-oriented development skills by learning to think in objects when faced with a design problem. This is evidenced by minimal use of class methods and data and proper use of abstraction, information hiding and encapsulation.

(Competency - Synthesis) Given a specification of requirements, analyze those requirements using CRC cards and scenarios. Select appropriate candidate objects representing the problem domain.

(Competency - Synthesis) Given a set of scenarios and CRC cards representing a customer problem, design an object-oriented solution and document that solution using the Unified Modeling Language (UML).

(Competency - Analysis) Apply standard design principles and patterns to a problem specification. Analyze a proposed design to determine its compliance with the standard principles (like open-closed, dependency inversion, law of Demeter) and make corrections as necessary.

(Competency - Application) Analyze a user interface to determine its usability using one of three standard techniques: Heuristic Evaluation, Cognitive Walkthrough or Think Aloud. Document your findings in a written report.

(Achievement - Synthesis) Given a problem specification, design, document and implement an object-oriented solution as a development team.

(Competency - Analysis) Demonstrate the ability to derive whitebox and blackbox tests from code or specifications. Document those tests in a basic test plan and implement those tests using an automated test environment such as JUnit.

Instructor

Bob Waters
Room 120 CoC
watersr@cc.gatech.edu

Office Hours: Tuesday 1000--1230.

Open Door Policy (If my door is open, it is OK to come in for help regardless of day).

Required Textbook

Recommended Text:

Agile Software Development, Principles, Patterns and Practices, Robert Martin, 0-13-597444-5

Optional:

Head-First Object Oriented Design and Analysis
web site: http://headfirstlabs.com/books/hfooad/

Head-First Design Patterns
web site: http://oreilly.com/catalog/9780596007126

Pro Android 2.0 (Spring only)

Head First JSP/Servlets (Summer only) http://proquest.safaribooksonline.com.prx.library.gatech.edu/book/web-development/jsp/9780596516680

These books are all available from the GT library Safari electronic collection.

Head First Object-Oriented Analysis and Design:http://proquestcombo.safaribooksonline.com.www.library.gatech.edu:2048/0596008678
Head First Design Patterns:http://proquestcombo.safaribooksonline.com.www.library.gatech.edu:2048/0596008678

Pro Android (Spring Semester Only):
http://proquest.safaribooksonline.com.www.library.gatech.edu:2048/book/programming/android/9781430226598

Killer Game Programming (Fall Semester Only(:
http://fivedots.coe.psu.ac.th/~ad/jg/

Electronic Readings

An Introduction to Object-Oriented Programming, Timothy Budd (**BUDD)**
Subversion (The Red Book)
Apache Ant

Architecture of Open Source Systems
Exploring CQRS and Event Sourcing : A journey into high scalability, availability, and maintainability with Windows Azure
J2EE Tutorial from Sun

Software Used In the Class

Java 6.0

Eclipse

Subclipse

Apache Ant

JUnit

Window Builder Pro

CodePro Analytix

Visual Paradigm

Android SDK

Collaboration Policy

Individual Assignments are just that, individual assignments. You may get general java help from others, but the code you write must be your own.

For the team milestones, teammates may freely use each other's code, but help between teams is limited to conceptual and general help. You may not just give other teams your code for their use. You may show them helpful classes, methods and packages however -- and show them how to use them. You just cannot hand them your code directly.

To use third party Java libraries that are not explicitly mentioned requires the permission of the instructor.

Use of copyrighted or offensive material in your projects is prohibited.

Grading

Grades are based on:

- 2 Exams (28)
  - Midterm - 8%
  - Final - 20%
- Class Quizzes - 5%
- 12 Group Project Milestones>Fall 2012 Project Milestones (67)
  - M1 Team Organization/Contracts/Project Management - 4%
  - M2 Domain Design -5%
  - M3 Application/Gui Design - 9%
  - M4 Subversion / Ant Lab - 5%
  - M5 Feature Set 1 - 5%
  - M6 Feature Set 2 - 5%
  - M7 Feature Set 3 - 5%

    - M8 Feature Set 4 - 5 %
    - M9 Feature Set 5 - 5%
    - M10 Feature Set 6 -5% (if we fall behind I may cut this and put percents in other slices)
    - M11 UI Evaluation - 7%
    - M12 Case Study Analysis - 7%
- Participation in class is highly encouraged and may be considered in borderline classes (e.g., Piazza postings, discussion in class, etc.)
- Participation in the project is mandatory. Student grades may be affected by participation depending on the team contract. See your team contract for details.
- Letter grades are assigned according to the usual convention (A=90+, B=80-89, C=70-79, D=60-69, F below 60), i.e.,**NOT** curved.
-
- Late assignments without a valid excuse, such as illness, will be graded this way:
    - 1 day: 90% (0-24 hours after t-square stops accepting)
    - 2 days: 75%
    - 3 days: 50%
    - 4 days: 25%
    - 5+ days: 0 points

Note: You may not receive extra credit points on an assignment that is Late

Attendance in lecture is not recorded, but expected.

All assignments are due at midnite on the due date, unless otherwise specified.

Email late assignments your grading TA. If you don't know who that is, email to the instructor.

Schedule

Date	Topic	Reading	Assignment
08/20	Course Introduction, Policies, Team Work, Team Work	Motivation and Teamwork Teamwork Guide	
08/22	Agile Development		
08/24	Team assignments and exercise	Chapter 1 and 3	M1 – Contract Released
08/27	OO History, Object Theory,		
08/29	OO Analysis, Identifying Objects		
08/31	CRC Cards, Use Cases, Responsibility-Driven Design, Stereotypes	Chapter 7 Using CRC Cards Use Cases Stereotypes msdn.microsoft.com/en-	M1 Due M2 Released

		us/library/ee658117.aspx	
09/03	SCHOOL HOLIDAY		
09/05	Software Architectures		
09/07	Responsibility-Driven Design, UML Class Diagrams		M3 Released
09/10	Responsibility-Driven Design, Sequence Diagrams	Appendix A	M2 Due M2 Meetings with TA due by Fri.
09/12	In-Class help session, Bob at Jury Duty	Handling Exceptional Conditions Presenter first coding	
09/14	Exception Handling and Contract design, Model-View-Controller		M3 Released M4 released
09/17	Build Files and Version ControlModel View Presenter in Java	Subversion Red Book Ant Manual	M3 Meetings with TA over next 7 days
09/19	Coding Standards and Code Reviews	Sun Coding Standards Best Practice for Peer Code Review	
09/21	Java GUI Data Binding and Graphics		M4 due M5 released
09/24	Design Principle – Single Responsibilty, Open-Closed, Liscov Substitution, Design Principle--	Chapter 8 Chapter 9	
09/26	Dependency Inversion, Interface Segregation	Chapter 10 Chapter 11, 12	Take home midterm released
09/28	Design Pattern –	Chapter 13, 14, 23	M6 Released

	Command, Template and Strategy, Delegation and Composite	www.laputan.org/mud/	M5 Due
10/01	Design Pattern – Façade, Mediator, Singleton, Monostate, Null Object, Factory	Chapter 15, 16, 17, 21	
10/03	Payroll Case Study	Chapter 18, 19	
10/05	Payroll Case Study		Take home midterm due
10/08	Testing, JUnit		
10/10	Testing, JUnit		
10/12	Packaging	Chapter 20, 22 http://www.infoq.com/presentations/code-organization-large-projects	M7 Due M8 Released
10/15	FALL BREAK		
10/17	User Interface Design		M6 Due M7 Released
10/19	User Interface Design		
10/22	User Interface Evaluation		
10/24	Design Reviews	Chapter 25, 26	M7 Due M8 Released
10/26	Design Review	Chapter 27	
10/29	User Interface Design		
10/31	User Interface Evaluation		M8 due M9 released
11/02	,Mark V Coffee Pot Design		
11/05	Mark V Coffee Pot Design		
11/07	Framework Design Weather Station Case Study Edutech Design Case Study	www.youtube.com/watch	M9 due M10 released M12 released
11/09	Design Critiques 9, 19, 3		
11/12	Design Critiques 29, 18, 17		
11/14	Design Critiques 4, 8, 28		M10 due) M11 released

11/16	Design Critiques 7, 23, 30		
11/19	Design Critiques 27, 31, 21		
11/21	Extra Credit Team Workday, Design Review		M11 Part 1 due
11/23	THANKGIVING HOLIDAY		
11/26	Security Design Guest Speaker - Counts as Quiz		M12 Due
11/28	Design Critiques 24, 35, 25, 33		
11/30	Web Design in Java		M11 due part 2
12/03	Virtual Machines/Garbage Collection		DEAD WEEK
12/05	Final Design Problem		DEAD WEEK
12/07	Team Final Prep		DEAD WEEK
12/10-12/14	FINALS WEEK		