CS2340 Fall 2013

Location

MWF 1205-1255

Clough Commons 144

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 0900--1200.

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

Required Textbook

Recommended Text:

Clean Code: A Handbook of Agile Software Craftsmanship
Robert Martin

Note: If you join ACM (cost $19), then this book is available on Safari for free.

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, <u>individual assignments</u>. 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)
  - Exam 1 - 8%
  - Exam 2 - 8%
  - Final - 12%
- Class Quizzes - 5%
- 12 Group Project Milestones>Fall 2013 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 - 6%

- o M12 Case Study Analysis - 8%
- 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:
  - o 1 day: 90% (0-24 hours after t-square stops accepting)
  - o 2 days: 75%
  - o 3 days: 50%
  - o 4 days: 25%
  - o 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 to your grading TA. If you don't know who that is, email to the instructor.

Schedule

Date	Topic	Reading	Assignment
08/19	Course Introduction, Policies, Team Work, Team Work	Motivation and Teamwork Teamwork Guide http://www.joelonsoftware.com/articles/fog0000000043.html	
08/21	Agile Development	Couch Potato/Hitchhiker http://www.codeproject.com/Articles/604417/Agile-software-development-methodologies-and-how-t	
08/23	Team assignments and exercise		M1 – Contract Released
08/26	Pivotal Tracker Intro to Design,	Chapter 1 BUDD On-line Chapters	
08/28	OO History, Object Theory		
08/30	OO Analysis, Identifying		M1 Due

	Objects CRC Cards, Use Cases, Responsibility-Driven Design, Stereotypes	Using CRC Cards Use Cases Stereotypes msdn.microsoft.com/en-us/library/ee658117.aspx	M2 Released
09/02	SCHOOL HOLIDAY		
09/04	Software Architectures, Trust Boundaries	Chapter 8, 11	
09/06	Responsibility-Driven Design, UML Class Diagrams	http://www.wirfs-brock.com/PDFs/A_Brief-Tour-of-RDD.pdf http://www.wirfs-brock.com/PDFs/Adding%20to%20Your%20Toolkit.pdf	M3 Released
09/09	UML Class Diagrams	http://www.ibm.com/developerworks/rational/library/content/RationalEdge/sep04/bell/	M2 Due M2 Meetings with TA due by Fri.
09/11	Responsibility-Driven Design, Sequence Diagrams	http://www.visual-paradigm.com/VPGallery/diagrams/Sequence.html	
09/13	Exception Handling and Contract design	Handling Exceptional Conditions Chapter 7 http://www.eiffel.com/developers/design_by_contract_in_detail.html	
09/16	Build Files and Version Control Model View Presenter in Java	Presenter first coding Subversion Red Book Ant Manual	
09/18	Coding Standards and Code Reviews	Sun Coding Standards Best Practice for Peer Code Review	
09/20	Java GUI Data Binding and Graphics	http://fivedots.coe.psu.ac.th/~ad/jg/	M3 Due .. Meetings with TA over next 7 days M4 released

Date	Topic	Reading	Milestones
09/23	Design Principle – Single Responsibilty, Open-Closed, Liscov Substitution, Design Principle--	Chapter 6, 10, 13 http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod	
09/25	Dependency Inversion, Interface Segregation		
09/27	Design Pattern – Command, Template and Strategy, Delegation and Composite	www.laputan.org/mud/	M4 due M5 released
09/30	Design Pattern – Façade, Mediator, Singleton, Monostate, Null Object, Factory		
10/02	Payroll Case Study		Take home midterm released
10/04	Payroll Case Study		M6 Released M5 Due
10/07	Testing, JUnit	Chapter 9	
10/09	Testing, JUnit		
10/11	Packaging	http://www.infoq.com/presentations/code-organization-large-projects	
10/14	Fall Break		M6 Due M7 Released

Date	Topic	Link	Milestone
10/16	User Interface Design		Take home midterm due
10/18	User Interface Design		
10/21	User Interface Evaluation		M7 Due M8 Released
10/23	Design Reviews		
10/25	Design Review Workday		
10/28	User Interface Evaluation		
10/30	User Interface Evaluation		
11/01	Mark V Coffee Pot Design		
11/04	Framework Design		M8 Due M9 Released
11/06	Weather Station Case Study Edutech Design Case Study	www.youtube.com/watch	
11/08	Design Critiques		
11/11	Design Critiques		M9 due M10 released M12 released
11/13	Design Critiques		

Date	Topic		Assignments
11/15	Design Critiques		
11/18	Design Critiques		M10 due) M11 released
11/20	Design Critiques		
11/22	Design Critiques		
11/25	Security Design Guest Speaker - Counts as Quiz		M11 Part 1 due M12 Due
11/27	Pre Break Topics		
11/29	Thanksgiving		M11 due part 2
12/02	Web Design in Java Virtual Machines/Garbage Collection		DEAD WEEK
12/04	Final Design Problem		DEAD WEEK
12/06	Team Final Prep		DEAD WEEK
12/09-12/13	FINALS WEEK		