

CIS 3238 Software Design

Course:	CIS3238
Course Title:	Software Design
Time:	MF 9:30 – 10:50 W (Lab) 9:00 – 10:50
Place:	MF TL305 W (Lab) CC 209
Instructor:	Paul Wolfgang
Instructor Phone:	215-204-5155
Office Hours:	MWF 1:00 – 1:50
Course Web Page:	www.cis.temple.edu/~wolfgang
Prerequisites:	C- or better in: CIS 2168 Data Structures CIS 3207 Introduction to System Programming and Operating Systems
Textbooks:	Flexible, Reliable Software Using Patterns and Agile Development Christensen CRC Press ISBN: 978-1-4200-9362-9 Object-Oriented Design & Patterns Horstmann John Wiley & Sons. ISBN: 978-0-471-74487-0
References	JUnit in Action, 2 nd Ed Tahchiev, Leme, Massol, & Gregory Manning Publications Co. ISBN: 9781935182023 Head First Design Patterns ¹ Freeman & Freeman O'Reilly ISBN: 978-0-596-00712-4 Test-Driven Development by Example Beck Addison-Wesley ISBN 978-321-14653-3 Growing Object-Oriented Software, Guided by Tests Freeman & Pryce Addison-Wesley

ISBN: 978-0-321-50362-6

Refactoring Improving the Design of Existing Code

Fowler

Addison-Wesley

ISBN: 978-0-201-48567-7

Pro Git

Scott Chacon

Apress

<http://git-scm.com/book>

¹ Available on-line through the Temple Library

**Course
Description**

Provides direct experience in the design, development, documentation, testing and maintenance of medium size software projects, in the use of modern software problem solving abstractions and solution patterns, and in the use of software development environments. This course is the capstone of the programming course sequence.

Course Goals:

To introduce the students to the following topics and to demonstrate their practical application.

- Graphic User Interface
- Access to Database
- Remote Procedure Calls (RPC)
- Extensible Markup Language (XML)
- Threading
- Object Oriented Design
- Unified Modeling Language (UML)
- Design Patterns
- Test Driven Development

To provide practical experience in using modern software development tools to perform the following tasks

- Project Management
- Version Control
- Build
- Test
- Issue Tracking

Provide experience working in small teams working on a project to modify an existing open-source program. Most Projects and code will start from or be inspired by:

- Sourceforge projects
- GitHub
- Apache Software Foundation projects

Grading:

- 10% Weekly quizzes - Brief in class quizzes to verify that students learn the material (generally on Fridays)
- 20% Labs - Small assignments intended to make student practice

with the concepts and tools presented

- 15% Midterm
- 30% Final Project - Six week long project done in small teams
- 25% Final Exam

Exam Dates:

- Midterm: Friday, Oct. 11
- Final: Friday, Dec. 13 8:00 – 10:00

Attendance Policy: Attendance is mandatory. Unexcused absence will result in reduction of final grade.

Labs

The labs are designed to reinforce the material presented in class. They form an important part of the learning process and make a significant contribution to your final grade. Labs are due one week after they are assigned. Late submissions are accepted, but at a reduced grade. Incorrect submissions that are on time will receive a reduced grade and can be re-submitted until correct to receive a slightly reduced grade. Completely skipping a lab submission will result in the reduction of your maximum possible final grade by one grade step for each skipped lab. Thus if you fail to submit 3 labs, the best grade you can get is a B.

Project

The goal of the project phase of this course is for students to gain experience in contributing to an open-source project. The nature of the contribution may be to contribute an enhancement or to fix an open problem. Students may work alone or in a team of up-to 3 students. While project work will be concentrated to the last half of the course, you need to select the project early and join the developer's mailing list. A written project proposal is due Monday, Sep. 16. This will contain:

- A one paragraph description of the overall project
- A one paragraph description of your proposed contribution
- A URL reference to the project.

Note: contributing to an open-source project is preferred. You may propose a stand-alone project, but if you do, it must be submitted as an open-source project.

Note: if you choose to contribute to an open-source project, be sure that you have down-loaded the source code and build the project.

Sources of open-source projects:

- Apache software foundation
 - <http://www.apache.org/>
- Source-Forge
 - <http://sourceforge.net/>
- Open Hatch

– <http://openhatch.org/>

Other Important Information

Disability disclosure:

Any student who has a need for accommodation based on the impact of a disability should contact me privately to discuss the specific situation as soon as possible. Contact Disability Resources and Services at 215-204-1280 in 100 Ritter Annex to coordinate reasonable accommodations for students with documented disabilities. (Temple University Policy and Procedures Manual)

Academic freedom:

Freedom to teach and freedom to learn are inseparable facets of academic freedom. The University has a policy on Student and Faculty and Academic Rights and Responsibilities (Policy #03.70.02) which can be accessed through the following link: http://policies.temple.edu/getdoc.asp?policy_no=03.70.02.

Academic Honesty

Academic cheating (such as plagiarism, copying during an exam, copying homework, stealing files and passwords, etc.) is strictly prohibited in this course. The penalty for the first offense will normally be an F in the course. A subsequent offense (in this or any other course) may also be referred to the University Disciplinary Committee.

No collusion what-so-ever during an exam will be tolerated. In particular not talking or other sharing of information (for example during open book exams) is permitted. Keep your eyes on YOUR paper.

IGNORANCE OF ACCEPTABLE GUIDELINES OF CONDUCT IS NO EXCUSE

http://policies.temple.edu/getdoc.asp?policy_no=03.70.12

Dates to Remember

First Day of Class:	Monday, August 26, 2013
Last Day to Drop:	Monday, Sep. 9, 2013
Project Proposals:	Monday, Sep. 16, 2013
Mid-term Exam	Friday, Oct. 11, 2013
Last Day to Withdraw:	Tuesday, Oct. 22, 2013
Last Day to makeup labs:	Wednesday, Dec. 4, 2013
Last Day of Class:	Wednesday, Dec. 4, 2013
Final exam	Friday, Dec. 13 (8:00 – 10:00)

Lecture and Lab Schedule

The lectures will be based on the Christensen and Horstmann texts and on other sources.

Week	Date (Monday)	Lecture Topics	Quiz/Exam	Lab
1	8/26/13	Introduction Test Driven Development Unit Testing Build Tools Christensen Chapter 5		Lab 0 Account Setup Practice with IDE, development tools, version control
2	9/2/13	Polymorphism, Interfaces and Subclasses, The Strategy Pattern Christensen Chapters 6 – 8 Horstmann Chapter 4	Quiz 1	Lab 1
3	9/9/13	Threading and Concurrency Horstmann Chapter 9	Quiz 2	Lab 2
4	9/16/13	PROJECT PROPOSALS DUE Parallel Processing	Quiz 3	Lab 3
5	9/23/13	XML RPC	Quiz 4	Lab 4
6	9/30/13	Advanced Unit Testing Mock Objects Test Stubs	Quiz 5	Lab 5
7	10/7/13	Review for Midterm	Midterm Exam	Lab Makeup
8	10/14/13	Initial Project Presentations		Project
9	10/21/13	Design Patterns The State Pattern Christensen Chapters 9 - 12 Horstmann Chapter 7	Quiz 6	Project
10	10/28/13	Compositional Design Christensen Chapters 15 - 18	Quiz 7	Project
11	11/4/13	Pattern Catalogue Christensen Chapters 19 - 29	Quiz 8	Project
12	11/11/13	Frameworks Christensen Chapters 30 – 32 Horstmann Chapter 8	Quiz 9	Project
13	11/18/13	Data Persistence and Database Access	Quiz 10	Project
14	11/25/13	Final Project Presentations (No lab this week)		
	12/2/13	Last Class (Review)		Last day to make-up labs.
	12/9/13		Final Exam	