EE363 – Software Components and Generic Programming

General Info

Instructor: Timothy C. Fanelli, CAMP 143
tfanelli@us.ibm.com
tfanelli@clarkson.edu

Mobile phone: 315-261-9711

Lecture: Monday, Wednesday and Friday
2:00 – 2:50pm

Lecture Room: Science Center 356
On the phone: 1-888-426-6840, access code: 31227078
On the web:
http://www.ibm.com/collaboration/meeting/join?id=FANELLI

Office Hours: Monday, Wednesday and Friday
12:00PM - 2:00PM

If I am not in my office, call my mobile # for office hours.

Catalog Description

A problem solving oriented course in software analysis and design with an emphasis on creating reusable software and software components in C++. Analysis and design of software components in the context of the C++ Standard Template Library

Working Description

This course is concerned with software design principles that foster creation of reusable software components. Topics include abstract data types, behavioral inheritance and subtyping, generics, interface design, dependency injection, and analysis of algorithmic behavior. Students will gain experience with software development best practices including design of test scenarios, unit testing, code reviews, refactoring and version control in the context of a modern integrated development environment.

Grading

Individual assignments will be given a letter grade, which correspond to the following grade-scale:

A+	A	A-	B+	B	B-	C+	C	D+	D	D-	F
97	93	90	87	84	81	77	74	68	60	50	0

Your final grade in this course will consist of 3 exams, two-hourly throughout the semester and a comprehensive written final, as well as several projects. Each hourly exam will be worth 15%, the final exam with 20%, and the projects worth 50%.

The final-average will be converted back to a letter grade, based on the following scale:

A (4.0)	B+ (3.5)	B (3.0)	C+ (2.5)	C (2.0)	D+ (1.5)	D (1.0)
>= 93	>= 87	>= 82	>= 77	>= 72	>= 67	>= 62

Where each lower-bound (e.g. “93”, “87”, etc) is representative of the lowest possible interpretation of that number. That is, “93,” here, should be interpreted as “93 point 0-repeating”.

Textbook

Head First Object Oriented Design and Analysis, O’Reilly Media
ISBN: 978-0-596-00867-3

Projects

This is a project-based course, and there will be several projects given throughout the semester. Projects will be posted on the website as they become available. Each project will have two deliverables, a written component and a source implementation. The weighting of each will be posted alongside in the project description.

The written component of each project will be graded on content as well as quality. We will discuss specifics as the projects are assigned, however be aware that “quality” includes spelling, grammar, and organization.

The source implementation components will be graded based on correctness of implementation, design, feature coverage, use of design patterns, and code-style and legibility. Your implementations will also be graded on how closely they match your written components.

Ethics

Professional ethics dictate that *all* assignments you turn in with your name on them represent your own work. You are allowed to work in teams, however, any two or more teams handing in identical (or overly similar) work will be investigated thoroughly.

The projects and assignments in this course will be sufficiently complex to warrant a non-trivial implementation, and as such, each implementation should be relatively unique. As I am relatively flexible on schedules and due-dates, and am willing to work with you individually to account for workload - I will have zero tolerance for copied or shoddy work.

Course Objectives

1. To teach students the basic principles of programming and software design
2. To develop student problem solving skills using programming as a tool
3. To teach students how to design and implement numerical algorithms
4. To familiarize students with the use of a modern software development environment