MET CS665

Software Design and Patterns

Class Organizations

Kia Teymourian, PhD

Course Description

- This course is an introduction to **software design and development, software *design patterns***
- We will learn about software design in general.
- This course covers the most common object-oriented software design patterns.
- A *design pattern* is a general reusable solution to a commonly occurring problem within a given contextin software design.
- Patterns are formalized best practices that the programmers should use and implement in the applications.

What will you get out of this course?

- You will have deeper knowledge about object-oriented programming
- You will understand general programming problems that programmers face frequently and know best practices that professional programmers use to solve them
- Your application will have better programming style and design
- Your code will be smaller and nicer.
- **This class improve your skills to become a professional Software Developer.**

Syllabus

Our Class is organized in 6 Modules

Module 1 - Introduction and Design Principles

Unified Modeling Language (UML), Class Diagram, Sequence, Use Case

Module 2 - Introduction to Design Patterns

Strategy pattern , Observer , Abstract Factory

Module 3 - Creational Design Patterns

Factory, Singleton and Prototype Patterns,

Module 4 - Structural Design Patterns

Facade, Decorator, Composite, Adapter, Flyweight and Proxy Patterns

Module 5 - Behavioral Design Patterns , Frameworks and Libraries

Observer, Visitor, State, Template and Strategy Patterns

Combination of design patterns, Frameworks and Libraries

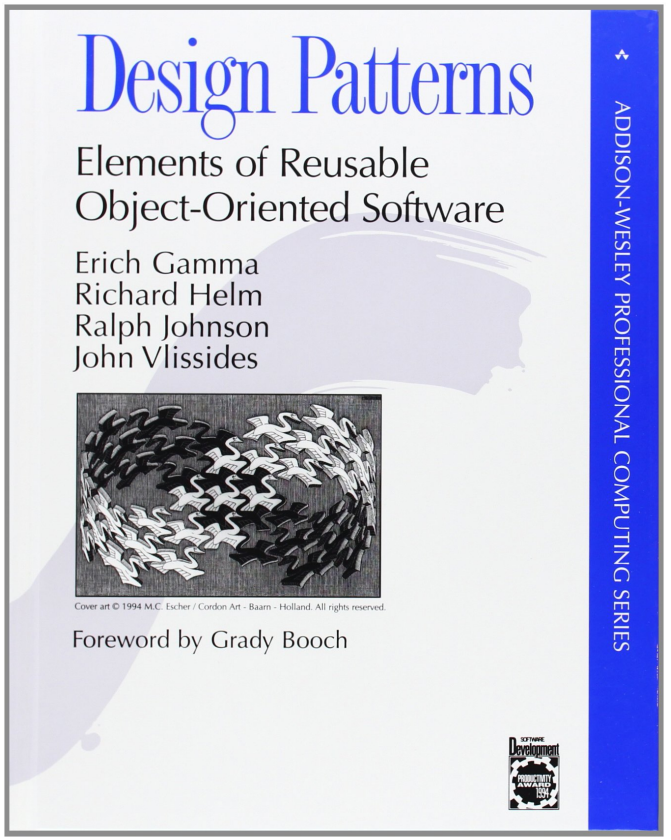
Module 6 - Refactoring and Code Cleaning

Prerequisite Courses

- Student should have solid background on object-oriented programming.
  - MET CS 341 or MET CS 342 (Data Structures with C++ or Java) is required for understanding the contents of this course.
  - MET CS 520 or MET CS 521
  - MET CS 526 (Data Structures and Algorithms) or MET CS 622 Advanced Programming Techniques
- The course is emphasizing the object-oriented design pattern concepts independent of programming language.
- All examples in class are using Java programs. But students have the choice of using either Java or C++ to implement the homework or project assignments.

Recommended Books

- **Design Patterns: Elements of Reusable Object-Oriented Software By Erich Gamma, Richard Helm, Ralph Johnson and John** Vlissides Addison-Wesley,1994



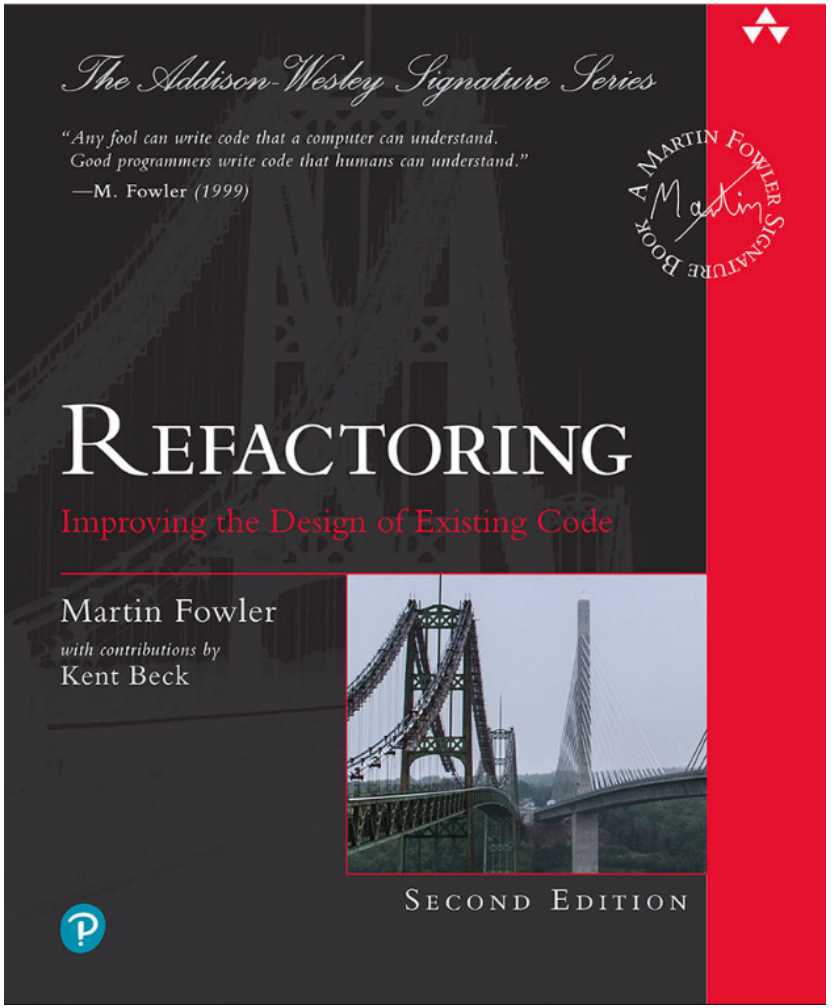
The most important book in this field.

Recommended Books

- Head First Design Patterns By Eric Freeman, Elisabeth Robson, Bert Bates, Kathy Sierra O'Reilly, 2004

Recommended Books

- Refactoring: Improving the Design of Existing Code. Martin Fowler



About me

Kia Teymourian

- Email: kiat@bu.edu
  - Include the class number in the subject of your email.

Tag Your Emails: *"[MET CS665] – Question about Assignment-1"*

- **Online Office Hour: Wednesdays 7:30-8:30 PM**

(send me an email before)

Academic Website: **http://www.teymourian.de**

Live Classes

- Zoom Live Class will be

Tuesdays

6:00 pm – 7:15 pm

Thursdays

6:00 pm – 7:30 pm

- We record all Live Classes

Recorded Video Lectures

For **each Module** you will find a set of recorded Video Lectures and slides.

You can find them on
"Instructor Resources" on Blackboard

Please watch all of the videos before Live Class.

We provide also some technical tutorials.

Grading Structure (cont'd)

6 Assignments	30%
6 Quizzes	20%
1 Term Project	20%
Final Exam	30%

A	95–100
A–	90–94.99
B+	86–89.99
B	81–85.99
B–	76–80.99
C+	71–75.99
C	66–70.99
C–	61–65
D	56–60
F	0–55

Grading Structure

- The **6 assignments** are focused on applying theory learned in the class to software applications.

- The **6 quizzes** will evaluate your understanding of concepts presented in the corresponding week's lecture.

- **One Term Project**. See Guidelines

- The **final exam** will be comprehensive and will cover material from the entire course.

Programming Assignment

- Assignments are real-world application scenarios
- You can learn a lot by developing and implementing the scenarios in CS665 assignments.
- We provide a brief description and the application use case.
- **Assignment Implementation Details:**

It is your task to make your own implementation assumptions, document them and implement your application based on your own assumptions. This will make **your solution to be different from each other**.

Programming Environment

- This class is not environment specific
- You only need to have Java Standard Endition (Java SE) installed on your system and have a normal text editor.
- Install Java and have a text editor
  - On windows: Java and Notepad++ or UltraEdit or Atom editor …
  - One Linux: Java and VI, VIM, nano or gedit or …
  - On Mac: Java and VI, VIM, nano or UltraEdit or …
- If you want you can download the Java Development Kit (JDK)
- You do not need to learn how to use an Integrated Development Environment (IDE) for this course.
- The programming tasks of this class is simple enough to work with a text editor.

Programming Style Guide

- Java
  - We follow the Google Java Style Guide
  - https://google.github.io/styleguide/javaguide.html

  Read the above document and follow its commandments

Programming Assignment

- You should implement in Java .
- Your program package should include a “read me” file with the name “README.md”

  It should describe, the purpose of program, how to compile and execute it.
- You should describe in README.md how to compile and run
  - In Java, you can use “**Maven**” to build your project
- You should document your code.
- We use Github and create private repositories for your assignments.

Class Public GitHub Repository

- We will add lots of examples in our Class Public Repository

https://github.com/kiat/MET-CS665/

- Examples and References.

Programming Template

- You should use a Maven Project Template

https://github.com/kiat/JavaProjectTemplate

- It is a Java Maven Project

Includes **Maven** setup **https://maven.apache.org/**

Includes Plugins for **Findbugs** **http://findbugs.sourceforge.net/**

Includes Plugins for **CheckStyle** **http://checkstyle.sourceforge.net/**

You can also use

- PMD Static Code Analyzer **https://pmd.github.io/**

Integrated Development Environment (IDE)

- It is a software application that helps programmer to work in a large software development project, like easier navigation in different package, search in source code files, source generation and code refactoring.

- If you are interested to us an IDE, we recommend to use
  - **Java**
    - **Eclipse** https://eclipse.org/
    - **InteliJ** https://www.jetbrains.com/idea/

- There are lots of other IDEs like **Netbeans**

GitHub Classroom Links

- We provide a **GitHub Classroom Link for each Assignments**

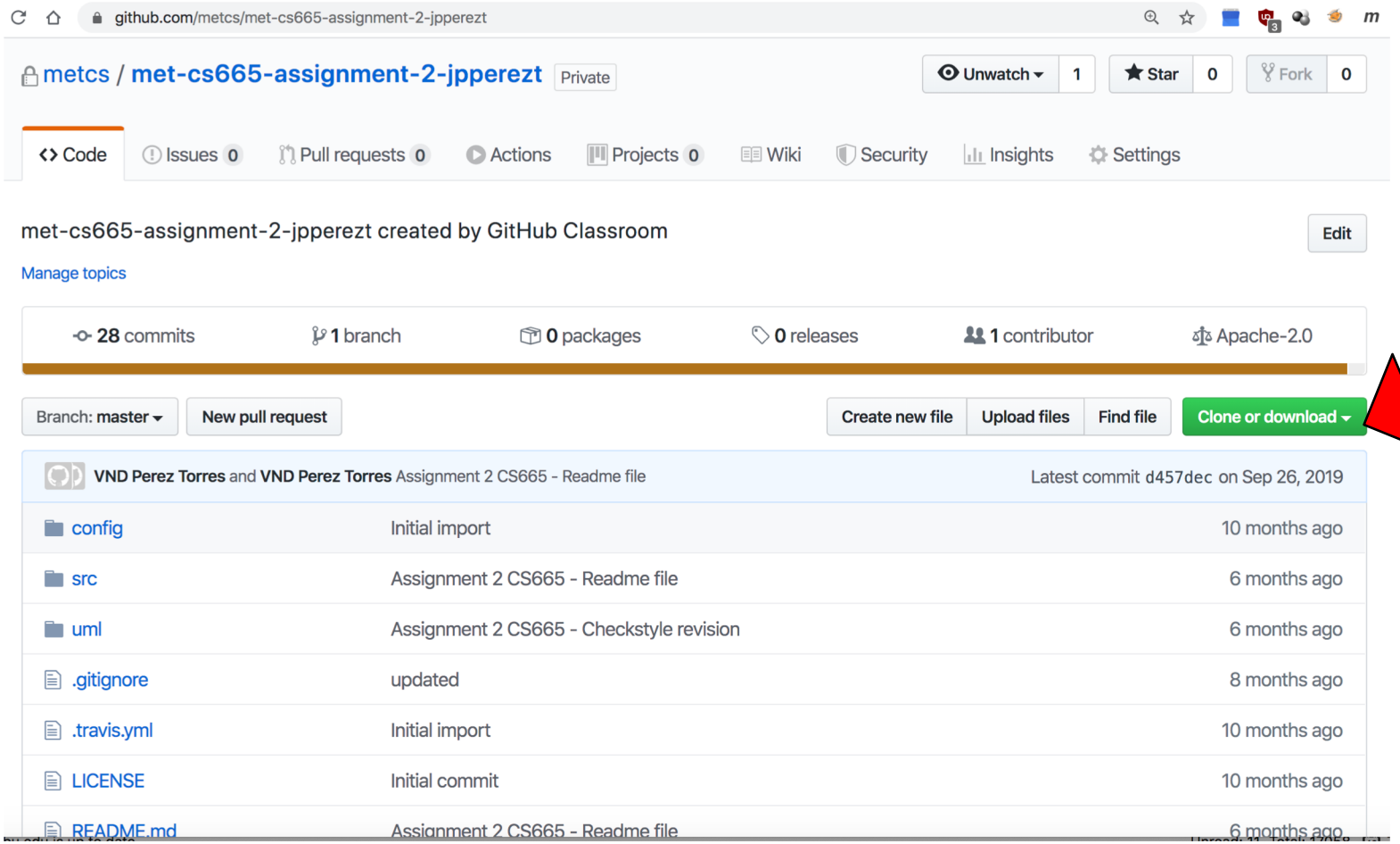
- **You should have your own GitHub Account.**
- Login to your own GitHub account and Click on the Assignment Link
- A Private Repository will be generated for your assignment
- Use the Private GitHub Repository to implement your assignment, and when finished create a ZIP file and upload it to Blackboard.

Assignment Submission

- Develop your assignment solution in your private repository on GitHub.
- Commit **and push all java source code and documents in your Github repository including README file, UML graphs , etc.**
- Do not push your Java Binaries (target) folder to github
- When you finished the assignment download a ZIP file from github and upload to Blackboard for our

Submit a Zip file to Blackboard

When you finished the assignment download a **ZIP file from Github and upload to Blackboard for grading.**



Download Here!

Assignment Evaluation Server

- Check your Assignment on our server
- You should see **”build successful”** and **”no bugs”**
- **http://server1.teymourian.info/metcs665/eval.html**

- **Good Example of a clean project build**
- http://server1.teymourian.info/cgi-bin/eval.sh?User=jpperezt&Nr=Assignment-2

Assignments

- Programming Assignments are overlapping between modules

- Assignment 1 → Modules 1
- Assignment 2 → Modules 1, and 2
- Assignment 3 → Modules 1, 2, 3
- Assignment 4 → Modules 1, 2, 3 and 4
- Assignment 5 → Modules 1 to 5
- Assignment 6 → Modules 1 to 6