Syllabus [Provisional]

Basic Information:

Course Title:	Software Engineering
Course Number:	CS-AD 209
Prerequisites:	CS-AD 104 - Computer Systems Organization CS-AD 105 - Algorithms
Lab:	None
Credit Hours:	4
Meets:	TR1120-1235, N-202
Professor of Record:	Michael Paik
Email:	mpaik@cs.nyu.edu
Office:	Sama Tower 1309A
Office Hours:	T1800-1900 and by appointment
Teaching Assistant:	None

Description: This course moves beyond the theory of computer science into the practice of building larger scale systems. Students are equipped to use industry-standard tools for the entire software lifecycle and learn to produce reusable, tested, quality code.

Topics include but are not limited to design and design patterns, refactoring, code optimization and performance, correctness and performance testing, threading and concurrency, security, user interfaces, web and mobile development, source code management.

Teaching Methodology: This course has two weekly 75 minute lectures. Lectures will be interactive, engaging students in the topic being discussed and encouraging students to ask and answer questions. Each lecture may be accompanied by a short programming or other assignment relating to the lecture material. Material will be presented using both slide decks and on whiteboards, as appropriate.

There are seven major modules, each of which will feed into the continuous improvement of work on the final project over the course of the semester. Instruction in the modules will be longitudinal and cover work to be done for server-side, web, and mobile technologies.

There will be two midterm exams, one in-class final exam, weekly or bi-weekly assignments and one final group project.

Examinations: Examinations will consist of True/False, Multiple Choice, Short Answer theory questions and practical programming exercises. Where possible, examinations will be administered on computers that have been disconnected from the Internet.

Assignments: Assignments will serve as constituent parts of the final project, and will be graded on the basis of correct application of concepts to the software development lifecycle. Students should be aware that failure to complete these assignments will result in penalties in both the Assignments portion of the grade as well as in the Final Project grade. There will be one to two assignments per major unit.

Final Project: Students are required to develop, in groups of at least 2, a substantial project utilizing the various skills and patterns covered in class. The project may be a game, web service or dynamic webpage, or server-side application, and must be approved by the instructor. 10% extra credit will be given if the application is judged to be useful to the wider NYUAD community and 5% if the application is judged to be useful on an individual academic or private basis. Note that these bonuses do not stack.

Grading Policy:

Participation:	10%
Assignments:	20%
Midterm Examinations:	10% each
Final Examination:	10%
Group Project:	40%

Intended Learning Outcomes:

Students who successfully complete this course will be able to:

- Develop a software system to solve a problem from scratch, including requirements analysis, design and prototyping, debugging, testing, and deployment.
- Understand management and deployment of live software with users
- Utilize critical software engineering concepts such as invariants, decoupling, abstraction, state machines, dependency, etc.
- Understand and use basic design and implementation patterns
- Set up and manage infrastructure for a medium-scale software engineering project
- Work collaboratively on a software development project
- Work with centralized and decentralized source code management tools

- Create unit tests providing complete source code coverage for automated regression testing

---

Lateness Policy: Late assignments will be penalized 10 points each day or part thereof they are late, weekends, holidays, and vacations included.

Extensions: *One* extension of up to one calendar week will be granted on an assignment, no questions asked. Any assignment which is turned in late using this extension is ineligible for extra credit, if any is available in the assignment.

Help: Students are **strongly** encouraged to make liberal use of email and office hours if they feel that they are falling behind. Special assistance is available by appointment.

Plagiarism and Academic Honesty Policy:

- Discussing assignment *concepts* is permitted, but all submitted assignments must be the student's own work.
- Copying all or part of another student's assignment or exam, or passing off another's work as one's own in the group project is prohibited. If detected, this will result in a grade of 0 on the relevant work for all parties involved. Repeated infractions will cause students involved to fail the course.
- Please note that a student is violating these guidelines, disciplinary actions will be implemented according to the official guidelines of the NYU Abu Dhabi Academic Integrity policy [1].

Student Conduct Policy:

- Students are expected to be on time to class. Lateness and absence will be counted against the participation component of the course grade. **Three** unexcused absences will result in a **letter grade deduction** in the final grade for the course (e.g. A to B); **five** unexcused absences will result in **failure**.
- Use of class time to read emails, social media, send or receive texts or calls on mobile phones, or browse the Internet is prohibited, and will be counted against the participation component of the course grade. Egregious violators will be ejected from class and such ejections will count as unexcused absences.
- Students are expected to be prepared by reading assigned material where applicable

[1] https://nyuad.nyu.edu/students/campus.life/policies/policy.academic.integrity.html

- The professor reserves the right to penalize repeated infractions with deductions from the final course grade.

Required Textbooks:

1. Gamma, Erich, Helm, Richard, Johnson, Ralph and Vlissides, John. *Design Patterns: Elements of Reusable Object-Oriented Software.* Addison Wesley, 1994, ISBN 978-0201633610 (aka *GoF* or *Gang of Four*, the reference standard)
2. Freeman, Elisabeth, Freeman, Eric, Bates, Bert, Sierra, Kathy, and Robson, Elisabeth. *Head First Design Patterns.* O'Reilly Media, 2004. ISBN 978-0596007126 (more accessible)
3. Summerfield, Mark. *Python in Practice.* QTrac, 2013. ISBN 978-0321905635
4. Hunt, Andrew and Thomas, David. *The Pragmatic Programmer: From Journeyman to Master.* Addison Wesley Longman, 2000. ISBN 978-0201616224
5. McConnell, Steve. *Code Complete, 2nd Edition.* Microsoft Press, 2004. ISBN 978-0735619678
6. Bloch, Joshua. *Effective Java, 2nd Edition.* Addison Wesley, 2008. ISBN 978-0321356680

Detailed Course Schedule: *(subject to adjustment)*

Module	Lecture	Date	Topic(s)
	1	28 Jan 2014	Introductory Material
Design	2	30 Jan 2014	Object-Oriented Design: Subclassing, Composition, Modular Dependency
	3	4 Feb 2014	Requirements and Specifications, Finite State Machines
	4	6 Feb 2014	Design Patterns
	5	11 Feb 2014	Concurrency, Threadsafety, Idempotence
Collaboration	6	13 Feb 2014	Collaborative Methods and Version Control
Human Factors	7	18 Feb 2014	Requirements Gathering, Behavior-Driven Development
Quality	8	20 Feb 2014	Unit Testing, Test-Driven Development, Continuous Integration, Blackbox and Glassbox Methods
	9	25 Feb 2014	Logging, Exception Handling, Debugging and Differential Diagnosis
	–	27 Feb 2014	Midterm I
	10	4 Mar 2014	Breaking Things, Load Testing, Fuzzing
Maintenance	11	6 Mar 2014	Refactoring, Documentation, Branching and Tagging, Regression Testing
Data	12	11 Mar 2014	Data Modeling
	13	13 Mar 2014	NoSQL, Object-Relational Mapping
	14	18 Mar 2014	SQL 1
	–	20, 25, 27 Mar 2014	No class (Spring Break)
	15	1 Apr 2014	SQL 2
Architecture	16	3 Apr 2014	Model-View-Controller
	17	8 Apr 2014	Service-Oriented Architectures
	–	10 Apr 2014	Midterm II
	18	15 Apr 2014	Middleware (Redis, Celery, MQ, etc.)
	19	17 Apr 2014	Asynchronous Design
Human Factors	20	22 Apr 2014	User Interfaces
	21	24 Apr 2014	Scope Creep, Requirements Churn, and User Acceptance Testing
Performance	22	29 Apr 2014	Load Balancing
	23	1 May 2014	Sharding
	24	6 May 2014	Static vs. Dynamic Content
	–	8 May 2014	Final
	25	13 May 2014	Presentation and Demo Skills
	–	15 May 2014	Group Presentations