



Department of Computer Science

College of Science and Mathematics
Montclair State University, Richardson Hall



Fall 2014

I. Course: CSIT 315-01 & 02 Software Engineering I

Pre requisite: CMPT 287

Instructor: Dr. Hubert A. Johnson Office: W314; Ph 655-7238
Office Hrs. M 4 – 4:50 am; T 3:00-3:50pm
R 12:00-12:50 pm.
Email: johnsonh@mail.montclair.edu

Text: Software Engineering, Theory and Practice 4th . ed., Shari L. Pfleeger, Prentice Hall Pub

Course Project: TEMPO Mobile Services

II. Objectives: This course:

- a) acquaints students with the basic concepts and major issues of software engineering
- b) introduces students to systematic approaches and techniques for requirement specification, analysis, design and software testing of high-quality complex software systems
- c) provides the experience of working in teams
- d) provides the experience of using a design tool (Rational Suite) in the design process

Upon completion of the course students should be able to:

- state the difference between a program and a software product
- analyze a problem, and identify and define computing requirements/resources appropriate to its solution
- state/identify the difference between programming and software engineering
- discuss the role of each phase in the software life cycle
- analyze, design, and develop test cases for testing a software system State and
- discuss various software development standards
- demonstrate an understanding of the concepts and principles of design
- apply design and development principles in the construction of software systems
- list and discuss categories, and examples, of data to collect for analysis
- discuss the different approaches to Standards and Certification
- demonstrate an ability to communicate effectively with a range of audiences
- produce high quality artifacts for a software system
- discuss software engineering ethics
- use CASE tools to aid in designing a software system
- demonstrate an ability to function effectively on a team to accomplish a common goal

III. Course Contents:

- The importance of Software Engineering
- Factors influencing Software Development
- Software Specification
- Software Analysis – Establish vocabulary of problem domain
- Design – Strategies, tools, patterns

- Issues (Specs, Design, Risks, Implementation, etc)
- Software Testing Techniques for analysis and design phases
- Project Management
- User Interface Design
- Cost Estimation
- Software Quality Management

IV. Requirements for completion of the course

- successful analysis, design, implementation, and testing a software system.
- Satisfactory achievement on tests
- Project Presentation:** Thur Dec 11, 2014

Test Dates (Tentative) Thur. Oct 9, Thur. Nov 13

Final Exam: Tuesday Dec 16; **Sec 01:** 10:15 am-12:15 pm; **Sec 02:** 3:15 – 5:15 pm

The grading is based on a point system according to the following:

Points	> 950	900-949	850-899	800=849	750-799	700-749	650-699	600-649	550-599	<550
Grade	A	A-	B+	B	B-	C+	C	C-	D	F

Points Allocation:

Tests	260
Individual Assignments	90
Projects	450
Final Exam	200

v. Attendance Policy:

Attendance is mandatory. Since you will be working in teams you need to be considerate of your team members. More than two unexcused absences will affect your final grade. For each unexcused absences beyond two, your grade will be dropped one grade point (A → A-, B → B-, etc). If you have more than five absences you will be dropped from the class. Make-up of missed work is your responsibility. **There are no make-up exam.**

Lateness:

Being late for class on more than four occasions will result in the lowering of your grade

vi. Academic Dishonesty (includes, but not limited to the following)

- The submission of work as one's own which was done by someone else
- Cheating on exams or tests: To give or receive assistance in any form during a test or an examination

There is severe penalty for engaging in academic dishonesty

NOTE:

- **The use of cell phones in the class is prohibited. All cell phones must be turned off or be placed in vibration mode**
- Check **Blackboard** at least three times per week

References

Booch, G. Software Engineering with Ada. Addison-Wesley 1997

R. Pressman, Software Engineering: A Practitioner's Approach, Addison Wesley

A. Davis, Software Requirements: Objects, functions, and States, Prentice-Hall Pub.

S. Schach, Object-Oriented and Classical Software Engineering, McGraw Hill

Sommerville, I. Software Engineering, Addison-Wesley, 1989

Davis, A. M. Software Requirements Analysis and Specification, Prentice-Hall 1990

Sherman, M. Software Engineering: Reliability, Development and management

S Bennett, S McRobb R Farmer, Object-Oriented Systems Analysis and Design using UML, McGraw-Hill, 2002.

P Stevens, R Pooley, Using UML - Software Engineering with Objects and Components, Addison Wesley, 2000.

M Fowler with K Scott, UML Distilled: Applying the Standard Object Modelling Language, Addison-Wesley, 2nd ed, 2000.

Overview

Software engineering (s.e) is concerned with long-term, large-scale programming projects. This course introduces the topic through lectures and by giving you a chance to help design, manage, and test a medium-sized project. The lectures and the group project will cover topics in s.e. management, problem specification and analysis, system design techniques, documentation, system testing and performance evaluation, reliable software, current design tools and techniques.

Homework

Most homework will be related to the course project (described below). There will also be reading assignments and occasional other exercises. There is an assignment write-up for each step of the project that includes a reading assignment. The readings should be done before you do the assignment. You are encouraged to read ahead as much as possible. Another course requirement is a weekly log of how much time you have spent on different activities related to the course. A simple log sheet is attached. You should turn in your log sheet each week at your last class meeting.

Canvas

Make use of Canvas in communicating with me. For each subsequent class meeting you are to post your questions/concerns regarding the previous class and/or project - difficulties, concept clarification, concept/idea you found particularly interesting, etc. **Be sure to check the Canvas assignments/announcements/discussion at least twice per week.**

Group Project

The class will be divided into teams of 3 (!!!) You are strongly encouraged to form your own teams (Regrouping by the instructor is a possibility.) The project for this semester will be to design a molecular docking software system for use by two professors in Bioinformatics. Each team will have regular weekly meetings (with a consultant). This is a semester-long project with major write-ups due at approximately two week intervals. There will also be some oral presentations, including a final demonstration of the project design.

Tests and Individual Projects

There will be two in-class exams and a final exam

Grades

Forty five percent (45%) of your grade will be based on the team project, documents and demonstration. You are graded on the quality of the work you produce, not on the number of hours a week you spend. Use your energy, and time, wisely. The rest of the grade will depend on individual tests, assignments, and contributions to the group project. It's very important that project assignments be turned in promptly, both to allow you time to 'complete' the tasks on schedule and allow time for evaluation of your work and for "suggestions!!!" to be made.

When in doubt, use as many categories as occur to you

Activity Log