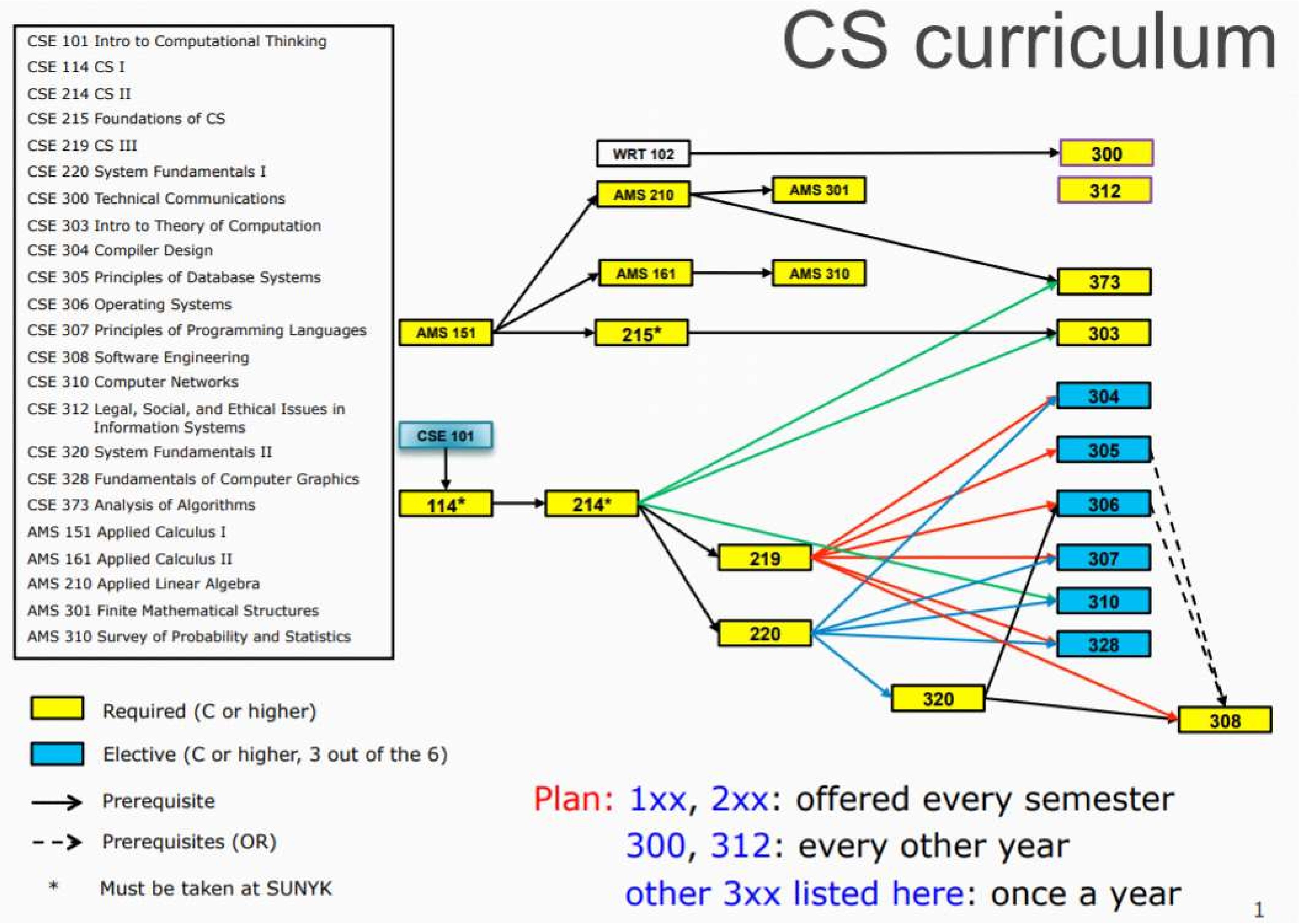
SUNY Korea
The State University of New York
한국뉴욕주립대학교

Spring 2019

CSE 308 : Software Engineering

LECTURE 0 – COURSE INTRODUCTION



CS curriculum

- CSE 101 Intro to Computational Thinking
- CSE 114 CS I
- CSE 214 CS II
- CSE 215 Foundations of CS
- CSE 219 CS III
- CSE 220 System Fundamentals I
- CSE 300 Technical Communications
- CSE 303 Intro to Theory of Computation
- CSE 304 Compiler Design
- CSE 305 Principles of Database Systems
- CSE 306 Operating Systems
- CSE 307 Principles of Programming Languages
- CSE 308 Software Engineering
- CSE 310 Computer Networks
- CSE 312 Legal, Social, and Ethical Issues in Information Systems
- CSE 320 System Fundamentals II
- CSE 328 Fundamentals of Computer Graphics
- CSE 373 Analysis of Algorithms
- AMS 151 Applied Calculus I
- AMS 161 Applied Calculus II
- AMS 210 Applied Linear Algebra
- AMS 301 Finite Mathematical Structures
- AMS 310 Survey of Probability and Statistics

🟨 Required (C or higher)

🟦 Elective (C or higher, 3 out of the 6)

→ Prerequisite

- -> Prerequisites (OR)

* Must be taken at SUNYK

Plan: 1xx, 2xx: offered every semester
300, 312: every other year
other 3xx listed here: once a year

Course Information

CSE 308 : Software Engineering

Course webpage: https://ppawar.github.io/CSE308-S19/index.html

Meetings: Lecture: Mon/Wed 3:30-4:50 PM

Place: B206

Prerequisites: C or higher: CSE 219 or CSE 260; CSE 320; CE305 or CE306; CSE major.

Staff

Instructor

- Pravin Pawar
- Office: B424
- Email: Pravin.pawar@sunykorea.ac.kr
- Phone: +82-032-626-1227
- Office Hours: *Tue/Thu 10:30 AM - 12:00 PM, Wed 5:00 PM - 6:30 PM*

Teaching Assistants

- Undergraduate (tutoring TA):

Course Overview

Introduces the basic concepts and modern tools and techniques of software engineering. Emphasizes the development of reliable and maintainable software via system requirements and specifications, software design methodologies including object-oriented design, implementation, integration, and testing; software project management; life-cycle documentation; software maintenance; and consideration of human factor issues.

The students work in teams of about four people to produce a significant piece of software during the entire semester. Each team produces a Specification Document, a Design Document, and a Test Plan. Each team gives an oral presentation of a design review and a formal demonstration of their project at the end of the semester.

Major Course Topics

Agile software development methodology

Waterfall model, together with its variations and improvements

The UML (Universal Modeling Language)

Requirements Analysis

Use Case Modeling

Object Modeling

Functional Modeling

Dynamic Modeling using State Diagrams, Sequence Diagrams, and Activity Diagrams

Specification Documents

Design Documents

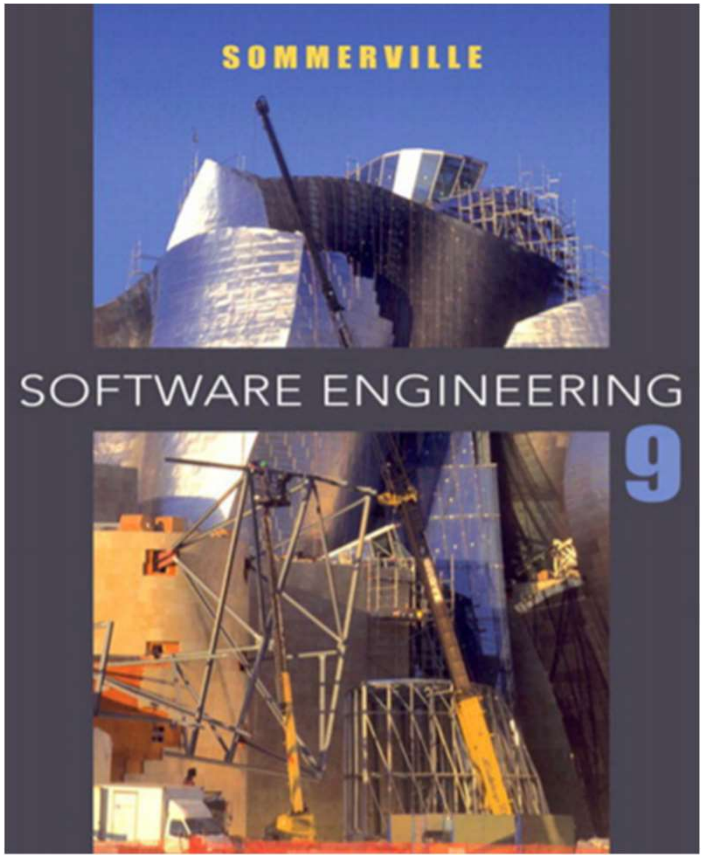
Testing -- Black Box, Glass Box and Test Plan Documents

Project Planning -- Pert Charts, Gantt Charts

Software Engineering tools

Oral and Written Communication

Textbook



The project

This semester the CSE308 project will be developing a mobile phone application of Commodity Ecology proposed by Dr. Mark D. Whitaker from the Department of Technology & Society at SUNY Korea. The United Nations acknowledged his leading work in understanding global commodity flows. See the following link for details:

https://academicimpact.un.org/content/commodity-ecology-initiative-facilitate-sustainable-development

The students will jointly work with Dr. Mark D. Whitaker (project customer) to understand the requirements of expected mobile application, approve the requirement specifications and develop this app to the level of satisfaction expected by him.

We will follow agile software development approach

Weekly scrum meetings

Project roles

For the group project, students will be divided into teams of four.

All students are expected to make contributions and the project benefits as a whole.

Note that all students are considered Software Engineers.

Additionally, students will be assigned the following roles:

- Lead programmer
- Project manager
- Lead designer
- Data designer

Lead Programmer

The lead programmer will be responsible for directing group decision-making regarding any technical aspects of the project, including design, implementation, and test decisions.

While not required, it would be helpful if this person has taken the Internet Commerce Programming and/or Principles of Database Systems courses.

Project Manager

A project manager drives the project, which means:

- Working with the lead programmer to direct the meetings (i.e. make the meeting agenda, assign action items, follow up on action items, take meeting minutes)
- Develop the Project Specifications
- Help divide project responsibilities
- Develop Gantt and Pert Charts for managing work measurement, forecasting, task breakdown, task assignment, and task dependencies
- Makes sure the project is progressing at the proper rate while using available resources (i.e. team members) efficiently.
- Will use project management software to organize the project.

Note that teams may decide to make this a floating position, rotating responsibility from one member to another.

At the end of the semester, full project management documentation must be provided specifying the full arc of the project.

Lead Designer

A lead designer is responsible for the look and feel, including interactivity, of the Web app.

This person should have a good taste.

Hopefully this person has some training in Web design or some knowledge of GUI Design Principles.

While not required, it would be helpful if this person has taken the Introduction to Web Design & Programming and/or User Interface Development courses.

Data Designer

This person is responsible for the data persisted to your site's database.

This means deciding what needs to be stored there and in what format to put it, and then making decisions regarding how to setup the database, how to update it, and how to make sure proper data is preserved and reset when necessary.

While not required, it would be helpful if this person has taken the Principles of Database Systems course.

Some Tips

Self-Learning is Key

- While we'll touch upon some of the topics needed for building Web apps, much will be left for team members to discover on their own.
- Early in the semester there will be an individual programming assignment that will introduce students to these technologies.
- Becoming skilled at self-learning is not easy, but is one of the best way to avoid career obsolescence.

Well Defined Contributions

- We will be using project management software this semester for managing all aspects of the development process.
- It will involve carefully defining each student's responsibilities and contributions.
- Students who fail to make expected contributions may receive a project grade lower than their teammates.

Team Programming

Late-work Policy

All assignments and milestones must be turned in by the due date and time.

- Any part of an assignment that’s late means the entire assignment is late.
- If your assignment is incomplete or not entirely working by the due date, turn in what you have to get some partial credit.

If you have an emergency situation, email me before the due date and I may be able to work something out

Bottom line: Plan ahead, start early!

Grading

Individual HW Assignment (10%):

- An individual programming assignment that uses all the needed Web technologies.
- 6 March – 18 March.

Quizzes (10%, 3 quizzes given, lowest grade dropped)

Midterm Exam (20%)

Group Project (60% - divided over 4 milestones and final project presentation):

- Software requirement specification along with wireframes (10%, due on 25 March)
- Software design document and technology choices justification (10%, due on 10 April)
- Alpha version and software test plan (10%, due on 29 April)
- Beta version and test results (due on (10%, due on 15 May)
- Final version along with deployment (20%, due on 5 June)

The final grade is based on the accumulated points from the individual HW assignment, mid-term exam and group project.

Re-Grading

To promote consistency of grading, questions and concerns about grading should be addressed first to the TA and then, if that does not resolve the issue, to the instructor.

You are welcome to contact the TA by email or come to his/her office hour. If you would like to speak with the TA in person, and have a schedule conflict with his/her office hour, you are welcome to make an appointment to meet the TA at another time.

For the assignments, quizzes and mid-term exams, request for re-grading must be made within one week from after the announcement of grades.

TA Assistance

TAs are available almost every day each week

- Schedule is forthcoming (posted on course web)
- In “CS Commons” (next to CSD office)

Come with specific questions and/or code with which you need help

- TAs strive to spend time with everyone that comes to a session so be courteous and share the TA’s attention

Electronics in Class

Cell phones should be put away during class

Laptops may be used during periods where you are asked to work on an exercise during class

Lecture slides are available on the course website for study before class

Talk to me after class if there's an issue with this policy

Disability

If you have a physical, psychological, medical or learning disability, please contact the Student Services and Career Team.

- Location: Academic Building A208
- Phone: 626-1190

The DSS will determine with you what accommodations, if any, are necessary and appropriate

All information and documentation of disability is confidential

How to Succeed in this Class

Attend class and be on time!
- Not all information is in my lecture notes or in the book
- I sometimes do in-class demos that emphasize non-obvious details

This is an introductory course, true, but we're going to cover a lot of ground and move quickly starting from scratch

The assigned work will take a lot of your time, so practice good time management

Read the reading assignments and review the lecture notes and try out example code
- Practice is the only way to become proficient at coding
- Very often your first, second, or third attempt at solving a problem will not be successful. It is **essential** that you give yourself enough time to try different ideas, taking breaks along the way!
- Those who write extra code for problems not assigned ("for fun") generally do best in this class
- Learning to code involves learning to read other people's code

Ask questions right away if confused. Ask in class, ask a TA, come to my office hours or send email. Don't stay confused and don't get behind!

Welcome and I hope you enjoy the class!