SCMP318 Software Development

James Skon

Spring 2018

Location: Hayes 203, Time: 12:10, Days: MWF

Office hours: 10-11 MWF, 2-3 M-F

Tutoring:

> *There are two ways of constructing a software design: One way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies. The first method is far more difficult. C. A. R. Hoare (1980 Turing Award Lecture)*

Course Learning Outcomes

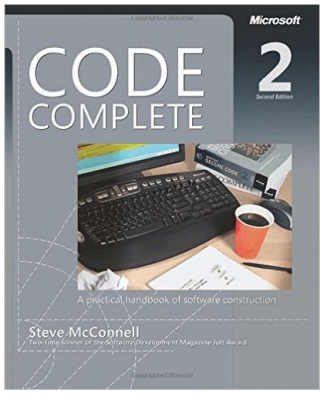
This course gives students experience designing, implementing, testing and debugging moderately complex systems of software components that collectively form a multilayer application. There will be an emphasis on crafting quality code, designing and implementing effective user interfaces, and building multicomponent architectures using a mix of off-the-self and custom code. Topics will include direct file I/O, inner-process and inter-system communication, multi-threading, and the synchronization of shared resources, web interfaces, data visualization and working with large data sets. For two projects students will work in project teams. Students will primarily use C++, but also will learn Javascript and other languages as needed. Prerequisite: SCMP 118 or permission of instructor.

Course Outcomes

At the completion of this course the students should

1. Exhibit best practices in creating code that is well structured and organized using object-oriented concepts.
2. Exhibit an understanding of quality use of identifier naming within code.
3. Be capable of providing appropriate internal documentation within code.
4. Understand and utilize proper use of internal barricading and error checking of values within a program.
5. Be capable of creating detailed requirement for a problem bending solved.
6. Be capable of creating architectural designs for multi-component software systems.
7. Be capable of collaboration in software development including pair-programming, peer design and code reviews.
8. Be capable of creating and using a MySQL database using SQL and phpmysql.
9. Write JavaScript code using HTML, CSS and jQuery.
10. Be able to create an HTML and JavaScript front end that communicate with a C++ program through an Apache2 web server.
11. Design and develop web based data visualization components and user interfaces that use quality metaphoric concepts.
12. Be able to produce JavaScript and/or C++ effectively processes XML documents.

Text

Code Complete, Second Edition; Steve McConnell; Microsoft Press; 2nd edition (June 19, 2004)	

Grading

Due Date: All assignments are due as specified in the grading table below.

Missing Lab Assignments: Labs are an important part of this class; the effort spent on them is a crucial part of the learning process. Failure to submit labs is unacceptable: students earning 0s on two labs cannot receive a grade higher than a B- for the course; students earning three 0s on labs will receive an automatic F for the course.

Collaboration and Academic Honesty: In order to facilitate learning, students are encouraged to discuss assignments amongst themselves. Copying a solution is not, however, the same as “discussing." A good rule of thumb is the “cup of coffee" rule. After discussing a problem, you should not take away any written record or notes of the discussion. Go have a cup of coffee or cocoa, and read the front page of the newspaper. If you can still re-create the problem solution afterward from memory, then you have learned something, and are not simply copying. (The “group work” are exempt from this, as they are intended to be done together.)

Category	%	Notes
Homework	15	Due by class time on day assigned. Not accepted Late
Quizzes	10	In class, at beginning. Weekly. No makeup without medical note.
Individual Project	30	Due by midnight on day due. 5% penalty for for up to 12 hours late. One assignment may be up to 24 hours late with no penalty with instructor notification. Projects have intermediate milestones and final complete project. Each milestones are typically worth 10-20 points, while the final grade is 100 points.
Team Project	30	Similar grading as above. All members are typically given the same grade unless there is evidence of certain team mambers.
Exams	15	There is a traditional individual midterm, while the final is a public group presentation on the final project.

Technical Topics

- Using the Linux Server
- C++ Iterators and algorithms
- Text Parsing
- Haspmap and building an inverted index
- Interprocess communication using FIFO class
- Client/Server architectures
- JavaScript, HTML and CSS
- JavaScript with timing intervals.
- AJAX and CGI communication
- XML Processing
- JQuery

- SQL
- SQL and C++
- Set Up a Node.js App

Notes

Tools

- cygwin - This is a tool to give you Linux software on a Windows system. A good way to get SSH.
- NetBeans
- Linux
- Notepad++ (Windows), How to install NppFTP Plug In to open and save file from server - link
- BBEdit (Mac - you can use the free version)
- PuTTY (Windows), ssh (Windows with cygwin)
- Filezilla - A tool for transfering files.
- X Windows for Mac
- TightVNC - X for WIndows
- ssh
- Make
- Git, Github
- Emacs

cslab.kenyon.edu links

- Monitor Active Processes (Enter filter like "cgi" to see matching processes)

Languages/Libraries

- C++
- HTML
- CSS
- Javascript
- Makefile
- XML
- Jquery
- HighCharts

Tutorials/Reference

- Using PuTTY to log in on from Windows
- Using SSH to log in on from Mac
- Using Notepad++ to edit and transfer files on Windows
- Using TextWrangler to edit and transfer files on Mac
- html & css
- Introduction to HTML
- Introduction to SQL Course
- Information on XMLHttpRequest Object
- AJAX introduction
- AJAX programming hints and suggestions
- Tips for using GitHub and git
- POSIX Threads Programming

- XML Parser for C++, Documentation
- 7 JavaScript Things I Wish I Knew Much Earlier In My Career
- Data Visualization with D3 course
- Using Node.js To Create Real-Time Web Applications
- Socket.io: let's go to real time!
- Socket: emit Cheat Sheet, Serving multiple file types.
-

Links

- Goverment Datasets
- Data driven literary analysis
- 11+ Killer Open Data Sources and Free Visualization Tools
- Big Data: 33 Brilliant And Free Data Sources For 2016
- What does this code do?
- Daily WTF - a how-not-to guide for developing software

Disability Statement

Kenyon College values diversity and recognizes disability as an aspect of diversity. Our shared goal is to create learning environments that are accessible, equitable, and inclusive. If you anticipate barriers related to the format, requirements, or assessments of this course, you are encouraged first to contact the office of Student Accessibility and Support Services (SASS) by emailing Erin Salva at salvae@kenyon.edu, then to meet with the instructor to discuss accommodation options or adaptations.

Schedule

Date	Topic	Reading / Info	Quiz	Slides	Assignment Due
08-31	Software Construction **Project 0**: Programming in the Linux environment	Chapter 1 Census Name Information Demo, ssh uname@cslab.kenyon.edu/SD18/NamesDemo		Software Construction	Student Information
09-03	Software Metaphors	Chapter 2 - Software Metaphors Linux Introduction PuTTY Project 0 Notepad++, BBEdit , EMACS HTTP & CSS, HTML Tutorial, Bootstrap introduction, Bootstrap Course.	Link	Metaphors Linux Slides linux command summery	
09-05	Name Data Demo, Inverted Index **Project 1, Part 1**: Shakespeare Index	Map STL c++ namesdemo.cpp Web Names Lookup NamesDemo code (github) Name Data Files - From US Census Data Filezilla - a tool for transferring files Project 1 Review			
09-07	Project Preparation HTML, CSS	Chapter 3.1-3.3 - Software Prerequisites Requirements Checklist CGI and AJAX Project 1	Link	Chapter 3	Project 0
09-10	Make Files, Bootstrap	Make Files, Make Tutorial, Bootstrap, BootStrap Course, BootStrap Tutorial	Link	Make Files Bootstrap	
09-12	**Makefiles** **Project 1, Part 2**: Simple Web Shakespeare	The Demo code for Name program The English stemmer example. C++ Web Programming		CGI and AJAX	HTTP & CSS
09-14	Web Programming with Ajax & Javascript	Ajax Tutorial for Beginners CGI and AJAX Name Data Web Program Link github Link		JavaScript jQuery	
09-17	Javascript and JQuery.	Javascript Tutorial, JQuery Tutorial, Learn JQuery, JavaScript & jQuery Tutorials	Link		Project 1, Part 1 Video to watch on Ajax JavaScript & jQuery Tutorials
09-	Project	Chapter 3.4-3.6 - More Prerequisites	Link	Chapter 3	

19	preparation Questions on projects				Bootstrap Meet with Professor for code review
09-21	**Project 1, Part 3**: Client/Server Web Shakespeare	Project 1-3 discussion Names with Client/Server			JavaScript and JQuery Tutorial Work
09-24	Key Construction Decisions Project 3-3 process communication	Chapter 4 - Key Construction Decisions Names with Client/Server Fifo's for communication	Link	Chapter 4 XML Overview	
09-26	Introduction to XML	XML Introduction , XML Tutorial - Review up to XML Attributes section before class for quiz.	Link	Introduction XML Part 1 Introduction XML Part 2 Introduction XML Part 3	Project 1, Part 2
09-28	**Project 2**: XML Lookup User Interface design	Project 1-3, Protocol Oveview , Tutorial: Parsing XML with JQuery MathML , Shakespeare , Bible , Quran XML Parser for C++ , Documentation Demo Software: /home/class/SoftDev/cppXMLAJAX/		Bible Example , Other examples	
10-01	Design in Construction Processing XML using jQuery	Chapter 5.1-5.3 Sample of C++ processing XML, CPP XML Example XML processing with jQuery, XML with bootstrap XML with Bible		Design in Programming	
10-03	Project 2 User Interfaces	Project 2: XML Project User Interface Design Basics Principles of User Interface Design User Interface Design Tips, Techniques, and Principles Interface Hall of Shame 		User Interface Design	Project 1, Part 3
10-05	Introduction to Git and Github	GIT Video - View for quiz , GitHub for beginners GIT HW	Link	GitHub.pdf Introduction to git and github	
10-08	**Visit to Gund Art Gallery**	Please meet at the Gund Gallery - be there before 12:10 if possible. Mapping Properties of Data in Visual Form			Project 2, Proposal
10-	**Project 3:** Data	visap2015_Cruz_WrongfullyRight.pdf - Please read			Project 2,

10	Visualization, metaphor, and visual communication.	before class, Visualizing empires decline 1, 2, Lisbon's Traffic 1, 2, 3, 4, An ecosystem of corporate, Embedded Space Visualizer,			Interface Design
10-12	**Midterm break**				
10-15	**Project 3 -** Project discussion and brainstorming.	Github example, Visualization Examples			
10-17	The use of Metaphor in visualization	Paper: On the role of metaphor in information visualization, What makes a good visualization. Review before class, and for quiz	Link	Metaphor and Visualization	GIT HW
10-19	More on Visualization, Group formation				Project 2, Complete
10-22	*Midterm Exam* - Study Guide	Chapters 1-5, User Interface Design, GIT	Link		
10-24	Team Presentations on Visualization Plan	Show mockups, explain goals.			Project 3 Proposal
10-26	MySQL and the World Database	SQL World Database, phpmyadmin, SQLTutorial, phpmyadmin Tutorial	Link	SQL	
10-29	Introduction to SQL, phpmyadmin, node.js	Learn SQL, Phone Number App Code, art.sql, PhoneApp			
10-31	**Project 4** : Phone App, MySql with C++	C++/MySQL tutorials, Socket.io cheat Sheet, Chat App, Chat Code			Project 3 Prototype SQL HW 1
11-02	Design Practices Project examples	node.js, node.js tutorials, socket.io, Socket.io: let's go to real time!, Chat Demo, NodeMySQL Example	link	Design in Programming	SQL HW 2-phpmyadmin
11-05	Working Classes	Chapter 6	Link	Class Design	Project 4 - Part 1
11-07	Demostrations	Chapter 7		Routine Design	Project 3 Complete
11-09	A visit to Gund Gallary.				Code Review Project 3
11-12	**Project 5**: Interactive two-user system with with Database	Project 5			

11-14	Automatic updating webpages Project 5 Group formation	Chat App , Chat Code, Live Monitoring Processes , code			Project 4 - Complete
11-16	Project Team work	Team meeting to work on project ideas			
Nov 17-24	**Thanksgiving Break**				
11-26	Project 5 Concept presentations	Be prepared to demonstrate and talk about your idea Chapter 7		Routine Design	Project 5 Part 1: Project concept,and team formation
11-28	Defensive Programming	Chapter 8	Link	Chapter 8	
11-30	Collaborative Programming	Chapter 21	Link	Chapter 21	
12-03	Group Work Day			Simple Gravity Example	Project 5 Part 2: Complete Project Design using Metaphoric concepts
12-05	Project 5 presentations	Present Architecture in class			Project 5 part 3: Architectural Design
12-07	Project 5 Team Work Day				
12-10	Personal Character	Chapter 33	Link	Chapter 33	
12-12	Developer Testing Project 5 Working prototype demos	Chapter 22		Chapter 22 Assessment Form	Project 5 Prototype Demo
12-14	Team work				Project 5 - Part 5 - complete system
12-20	8:30-11:30am	Final Presentation Project 5		Evaluation Form	Final Presentation

					Personal Statement

Software project grading rubric

Criteria	Excellent	Acceptable	Unacceptable
Documented & Maintainable *(The program is well-documented with appropriate names and comments making it easy to understand.)*	• all naming conventions are followed • both in-line and header comments are included and clearly explain the what the code accomplishes and how • white space is used well	• most naming conventions are followed • some comments are confusing or missing • white space is used well in most places	• poor or no use of naming conventions • too few or too many comments are used and they are unclear or inaccurate • poor use of white space
Adaptable & Reusable *(The program is modular, using abstraction well and any limitations are clearly specified.)*	• all interfaces between objects are clear • appropriate utility functions are used and well-documented • most code can be reused	• most object interfaces are clear • some appropriate utility functions are used and documented • some code can be reused	• poor object interface definitions • few or no utility functions • no code can be reused
Robust & Correct *(The program provides the correct output for all possible input.)*	• the program works completely as expected • the output is displayed to specification for all valid input • the program responds appropriately for all invalid input	• the program works as expected for most input • there may be minor errors in output formatting for valid input • not all invalid input is handled reasonably	• the program does not produce correct output for even the sample input • the program fails to handle invalid input • exceptions are not caught
Efficient & Elegant *(The program uses both time and space on the computer effectively, without losing source code clarity.)*	• no extra variables or definitions are used • the code is small, efficient yet still easily understood	• extra variables do not make the code harder to understand • brute-force problem solving approach	• extra variables are pervasive and confusing • the code is unnecessarily long and patched together

	25-20%	19-11%	10-0%

I	Attachment	History	Action	Size	Date	Who	Comment
	0809.0884.pdf	r1	manage	524.2 K	2018-10-13 - 01:34	JimSkon	
	art.sql	r2 r1	manage	11699.8 K	2018-10-26 - 03:35	JimSkon	

Topic revision: r55 - 2019-05-22 - JimSkon