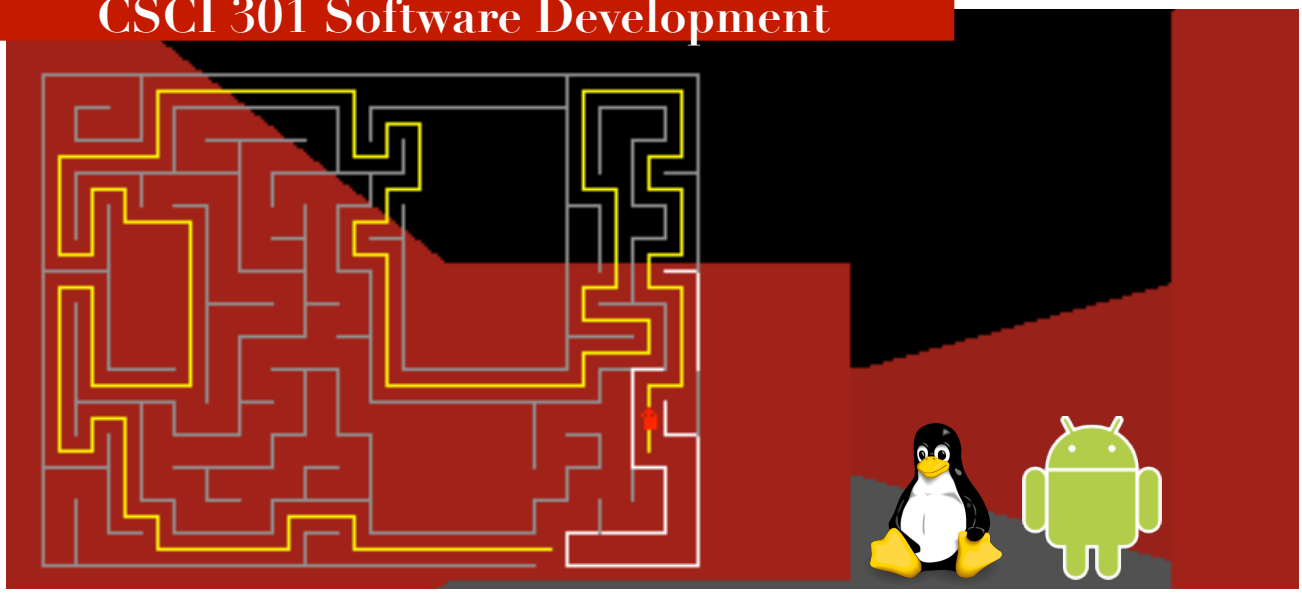
CSCI 301 Software Development



Pictures of Linux™ mascot &Android™ logo from Wikipedia.org
Screen capture taken from Paul Falstad's maze application

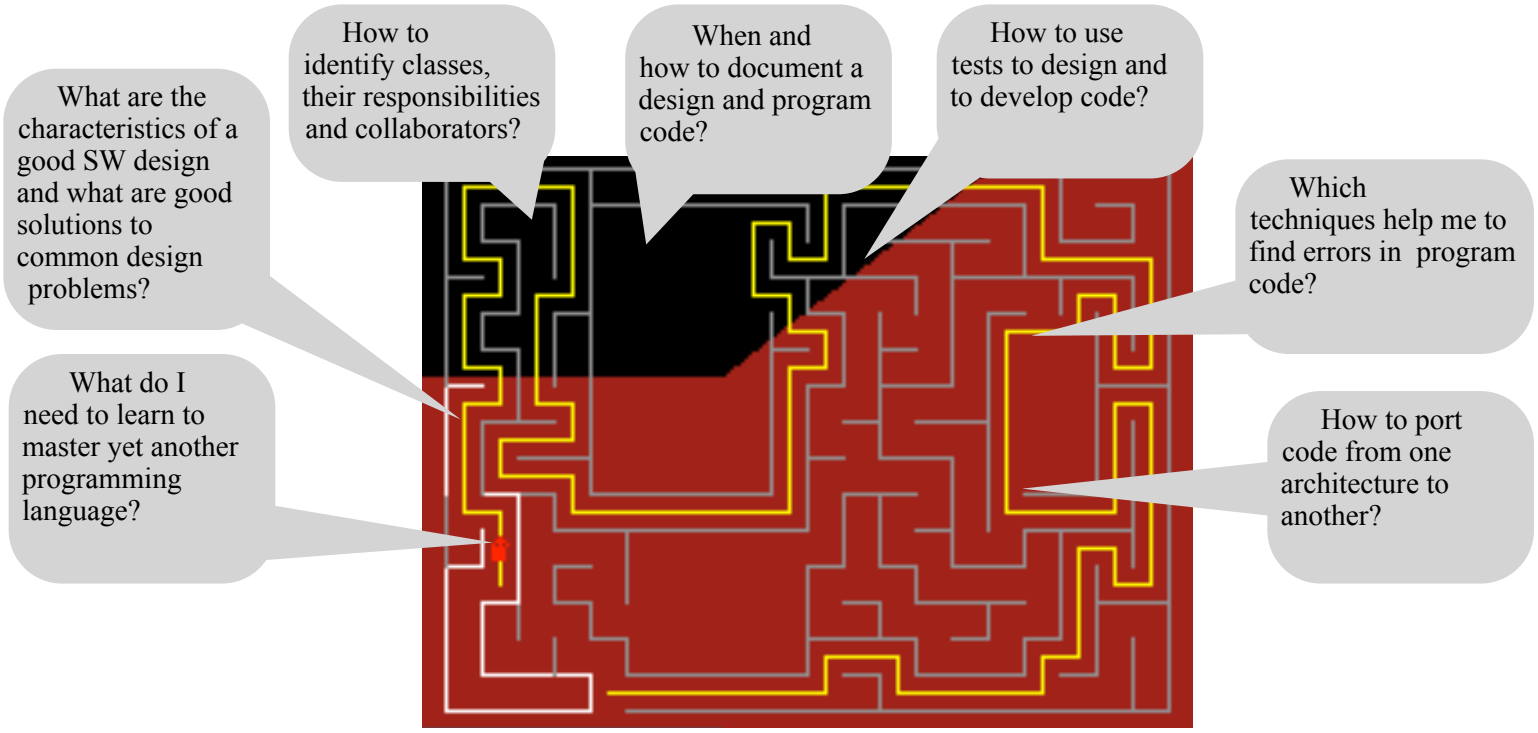
CSCI 301 Software Development

3 Credits, Prerequisite: CSCI 241, a working knowledge in Java

From the catalogue: "An introduction to principled software development, emphasizing design at the module level as well as tools and techniques. Topics include object-oriented class design and implementation, abstraction techniques, debugging techniques, defensive programming, development and analysis tools, and testing. Emphasizes the role of the individual programmers in large software development projects."

Content: The first half of the course will cover object-oriented programming in Java, software patterns and a test-driven software development. It addresses the particular challenges that a software developer faces when a project scales to the size of a real-world multi-person long-term project of considerable size and complexity. A selection of software development tools (Eclipse as an IDE, Junit for unit testing, Eclemma for code coverage, Git for versioning) will be introduced and used in project assignments.

Our pathway through the Maze of Knowledge & Skills for CS 301



Software Development Techniques
- Object oriented design
- Patterns for software design
- Test Driven Design
- Separation of user interface layout design and control flow

Programming Techniques
- Java programming (Interfaces, Inheritance, File I/O)
- Multithreading
- GUI programming (Java, Android)
- Message passing (Android Intents)
- Database access (Android SQLite)

Development Tools
- Integrated development environment (Eclipse, Android Studio)
- Versioning systems (Git)
- Documentation (Javadoc)
- Automated testing (Junit)
- Static code analysis (Spotbugs)
- Emulator (Android)
- Debugger

The second half of the course will cover software development for mobile applications, in particular object-oriented programming in Java for App development on Android. The software development will use the Android emulator that interacts with the Android Studio IDE such that no additional hardware (no mobile phone) is required.

While the first project is small scale and special to get started with Java. All other project assignments will build and extend on a single overall project through a series of refinements, extensions and adjustments. The starting point is an existing maze application that is implemented in Java and allows a user to explore a randomly generated maze from a first-person perspective as well as from the top (see screen shot above and below). We will extend the functionality of the existing code base and subsequently transform it into a mobile app running on Android platforms.

Instructor

Peter Kemper
104A McGlothlin-Street Hall
Tel: +1 757 221-3462
Email: kemper@cs.wm.edu

Required book

There is no particular book that you are required to buy.

Recommended reading

Steve McConnell, Code Complete, 2004. Online access via SWEM library.

Cay Horstmann, Object Oriented Design & Patterns, 2nd edition, Wiley.

Android: developer.android.com

Required work and grading

This class adheres to a learning-by-doing approach, so the required work will mean programming in most cases.

Projects (80%): There will be a series of usually biweekly project assignments that add on to each other. The assignments require significant effort and time and will give in total 80% of the final grade. While the first project is separate, all others are related and step by step will build a module of useful new functionality into an existing real world software system. Projects will be done on your own, as a formal matter of honor. The operative rule is that you may consult with your classmates on general issues about an assignment, share references to various sources of information, discuss possible solutions and coding issues, but you write your own code for your solution and you do not copy other peoples code.

Tests & Final Exam (20%): There will be a series of online tests (4 tests, 2.5% each, 10% in total) and a final exam (10%) posted on blackboard. The final exam duration is 2 hours.

Late work policy

Assignments come with a hard and final deadline. An assignment that you hand in before the deadline will be considered and graded. An assignment that you hand in after the deadline will NOT be considered and NOT graded.

In order to accommodate for unusual circumstances that may occasionally happen in life, each student will be granted once a one-week deadline extension for a single project assignment as an exception to this rule. The extension is one-week regardless of its reason to provide sufficient time and flexibility to finish the assignment together with other assignments from other classes.

So it is highly recommended to plan ahead and work with personal deadlines that include a little safety buffer for your own sanity. Due dates are typically set to 11.59 pm to discourage students from planning for all-nighters, which are considered unproductive and unhealthy. Deadlines will be set well in advance and are shared on BB, specifically with a shared Google calendar that you can link into your own.

Students who need accommodation

I am happy to accommodate any student with disabilities or additional needs so they can be successful in this class. Please contact me privately and, at your convenience, we will set up an appointment to discuss and prepare appropriate accommodations.

Absence from classes

If students can not attend classes for an acceptable reason (a positive Covid test or symptoms may serve as an example), they are expected to notify the instructor, participate remotely to the extend their health situation or other circumstances permit by watching recording lectures and work through instructional material or links provided on the Course site on Blackboard, and work on project assignments and tests.

Information Dissemination

I will maintain a blackboard course and use Piazza as a communication platform.