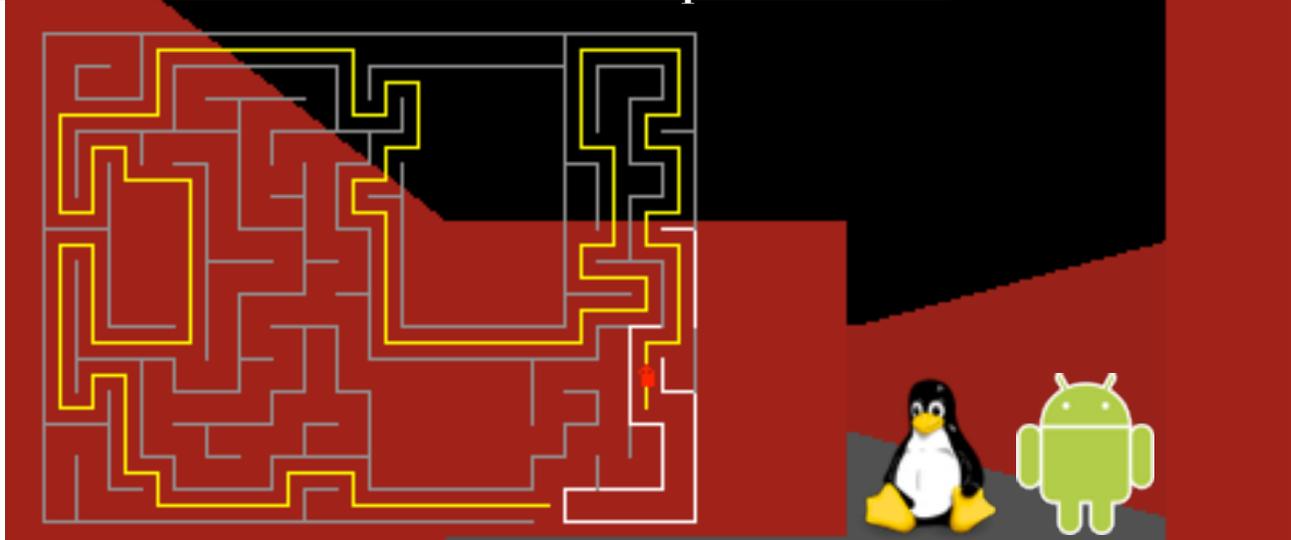


CSCI 301 Software Development



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."

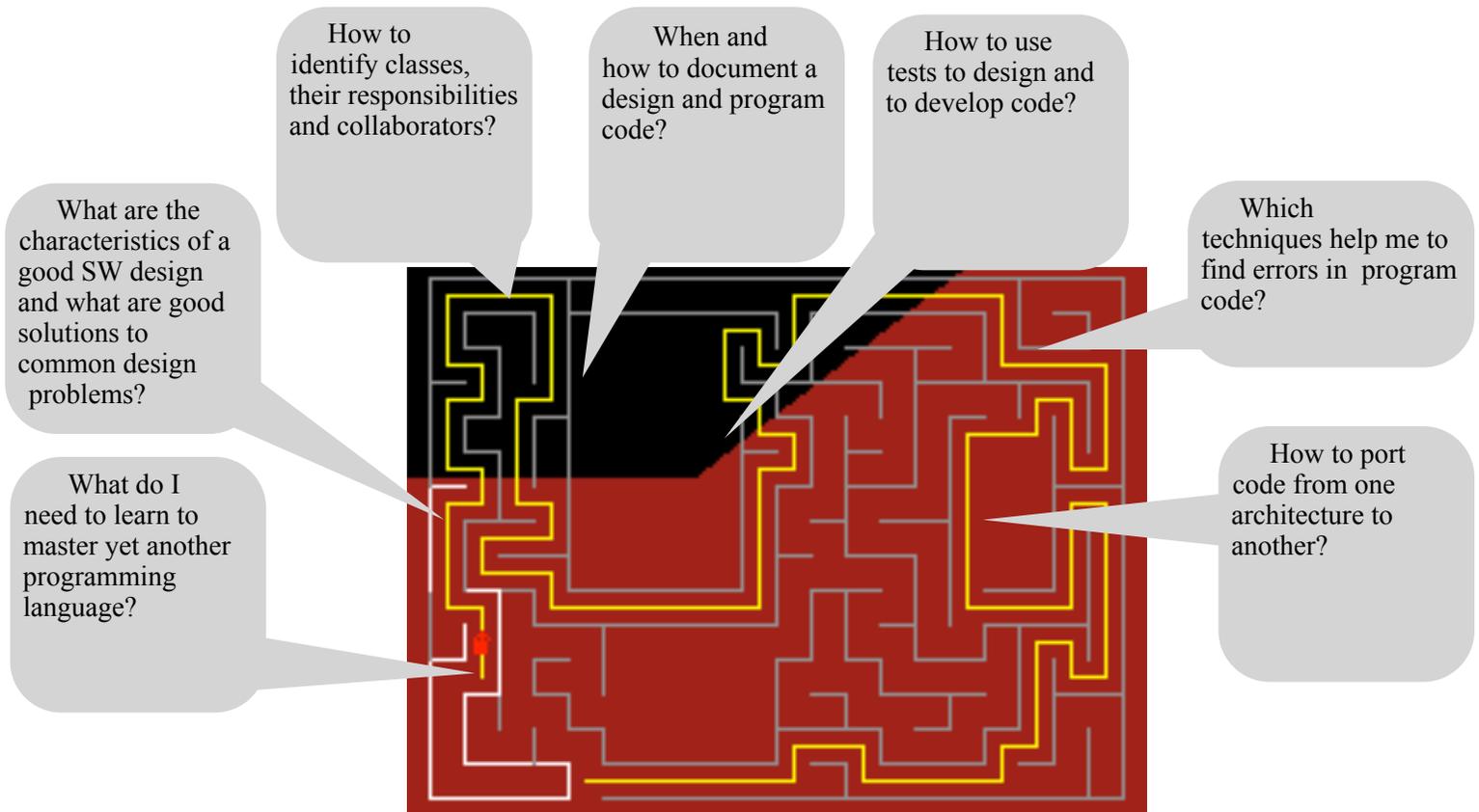
Content: The first half of the course will cover object-oriented programming in Java, software patterns and a test-driven software development. It addresses the particular challenges that a software developer faces when a project scales to the size of a real-world multi-person long-term project of considerable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

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 Eclipse IDE such that no additional hardware (no mobile phone) is required.

A single overall project will be developed through a series of individual homework and project assignments. 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). We will extend the functionality of the existing code base and subsequently transform it into a mobile app running on Android platforms.

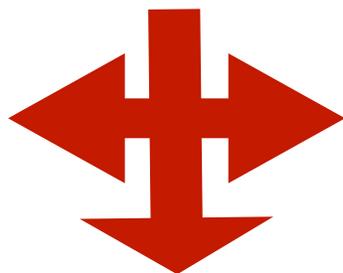
CSCI 301 Software Development

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)
- Versioning systems (Subversion)
- Documentation (Javadoc)
- Automated testing (JUnit)
- Static code analysis (Findbugs)
- Emulator (Android)
- Debugger

CSCI 301 Software Development

Instructor

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

Grader: Yongsen Ma, Hongyang Zhao

Where and When

Class: Section 1: 9.00-9.50 am, MWF, McGlothlin Street Hall 020

Section 2: 11.00-11.50 am, MWF, McGlothlin Street Hall 020

Office hours: Tuesday 1-3 pm, Friday 1-3 pm, and other hours by appointment

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. (One hard copy available for short term at SWEM library)

Mark Murphy, The Busy Coder's Guide to Android Development, commonsware.com (Mark has donated coupons for his book in the past and will hopefully do so for this class again.)

Required work and grading

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

Projects and Homework assignments (80%): There will be a series of usually weekly homework and projects assignments that add on to each other. The assignments require significant effort and time and will give in total 80% of the final grade. The projects 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, but own code remains private and is not shared. There is the option that two students join for [pair / partner programming](#) which needs to be approved by the professor before the project is assigned. Pair programming comes with an extra set of rules to obey but also with the potential of learning more and performing better in class.

Tests and final exam (20%): There will a series of online tests and a final exam which will give 20% in total for the final grade.

Active class attendance: This is a course that requires you to be present and to actively participate.

CSCI 301 Software Development

Late work policy

All assignments come with a hard deadline. An assignment that you hand in after the deadline will NOT be considered and NOT graded. In order to accommodate for unusual circumstance that may occasionally happen in life, each student will be granted one exception to this rule. So it is highly recommended to plan ahead and work with personal deadlines that include a little safety buffer for your own sanity. Deadlines will be set well in advance.

Attendance

It is expected that students attend all classes.

Students who need accommodation

Please see me after class or send email to set up a brief meeting.

Information Dissemination

I will maintain a blackboard course and additional material on a class wiki in support of the course. I will use Piazza as a communication platform.