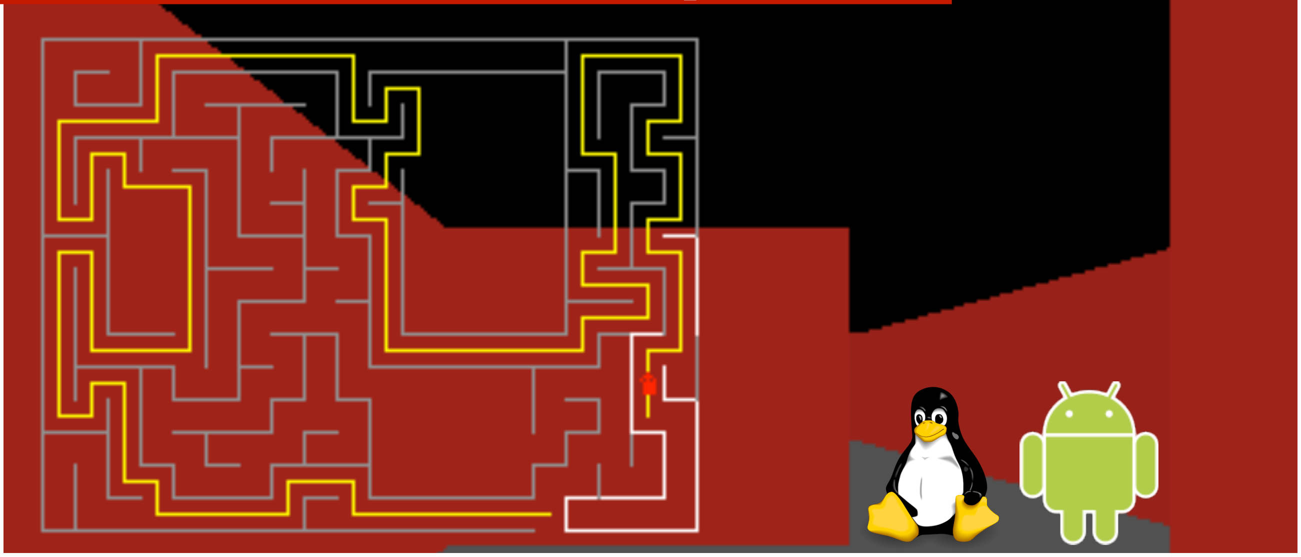
CSCI 301 Software Development



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."

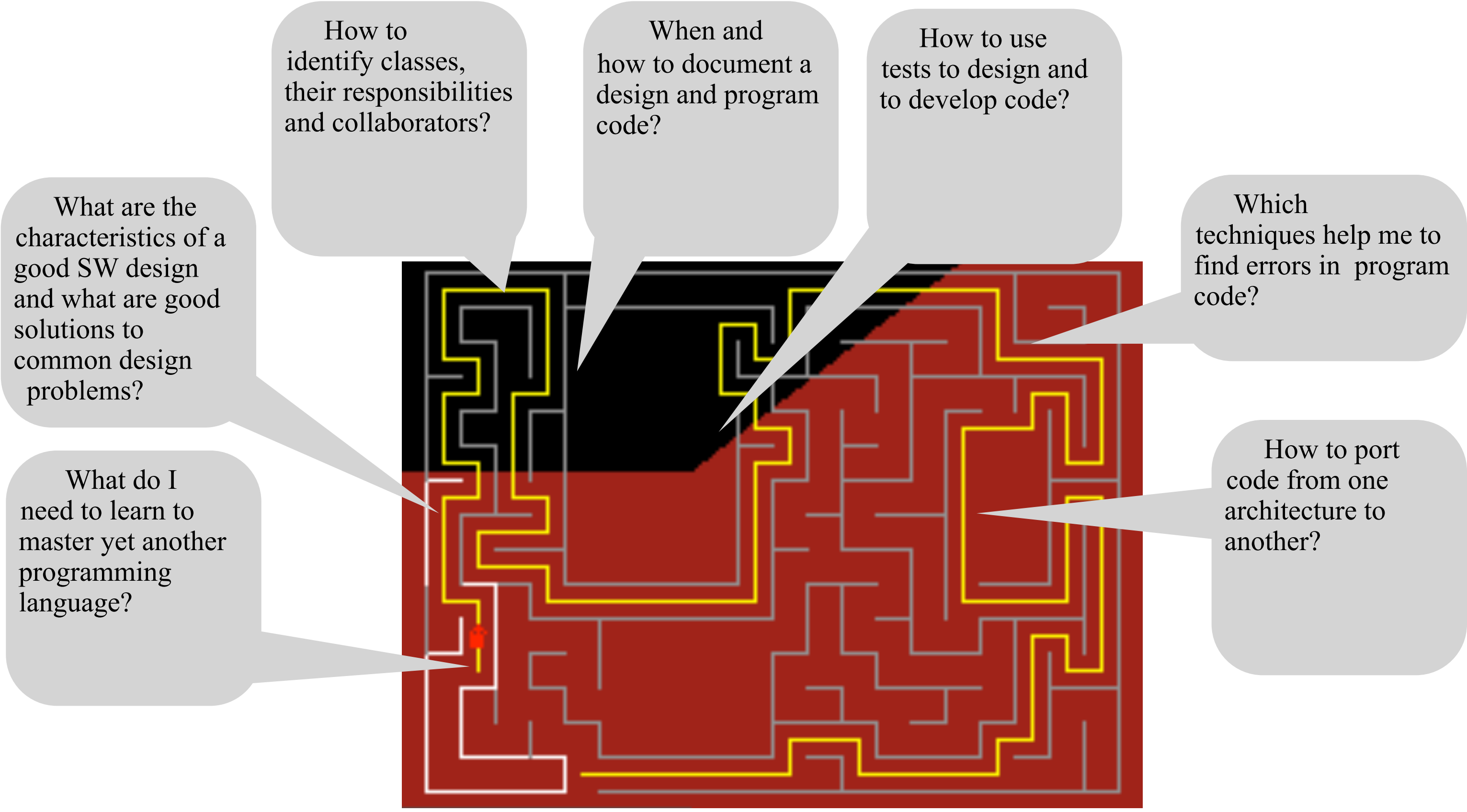
Flipped classroom: The pedagogical format of this course follows a blended learning approach that requires students to work through course materials and assessments before each individual meeting in class. During class time, learning activities focus on problem solving and learning to apply the content covered in the course materials. This format is used in Fall 2014 for the first time.

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.

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



How to identify classes, their responsibilities and collaborators?

When and how to document a design and program code?

How to use tests to design and to develop code?

What are the characteristics of a good SW design and what are good solutions to common design problems?

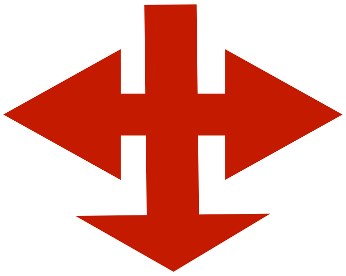
Which techniques help me to find errors in program code?

What do I need to learn to master yet another programming language?

How to port code from one architecture to another?

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: tbd

Where and When

Class: 11.00-11.50 am, MWF, MCG 020

Office hours: Monday 2-4 pm, Thursday 1-3 pm, and other hours by appointment

Required book

tbd.

Recommended reading

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

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

Edward Crookshanks, Practical Software Development Techniques, 3rd edition, 2012.

Crookshanks' book introduces techniques and tools used in software development (and CS 301) in a very pragmatic and brief manner.

Required work and grading

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

Problem solving (30%): There will be a number of assignments to work on in class and possibly finish at home. Results will be graded and the average grade over all problem solving assignments will give 30% of the final grade. These assignments will be done on your own, as a formal matter of honor.

Projects (50%): There will be four to five projects assignments that require significant effort and time and will give in total 50% 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 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.

Exams (20%): There will be a midterm and final exam. Exam results will give 20% of the final grade.

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

Late work policy

Assignments come with a submission deadline and a drop out deadline. The submission deadline is when you are supposed to hand in your results. Time management by submission deadlines will always keep you on the safe side. The drop out deadline is later. An assignment that you hand in before the drop out deadline will be considered and graded. An assignment that you hand in after the drop out deadline will NOT be considered and NOT graded.
So it is highly recommended to plan ahead and work with submission deadlines. 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 set of web pages beneath http://www.cs.wm.edu/~kemper/cs301/ in support of the course. They should be considered official components of the course, and you should check them on a daily basis.