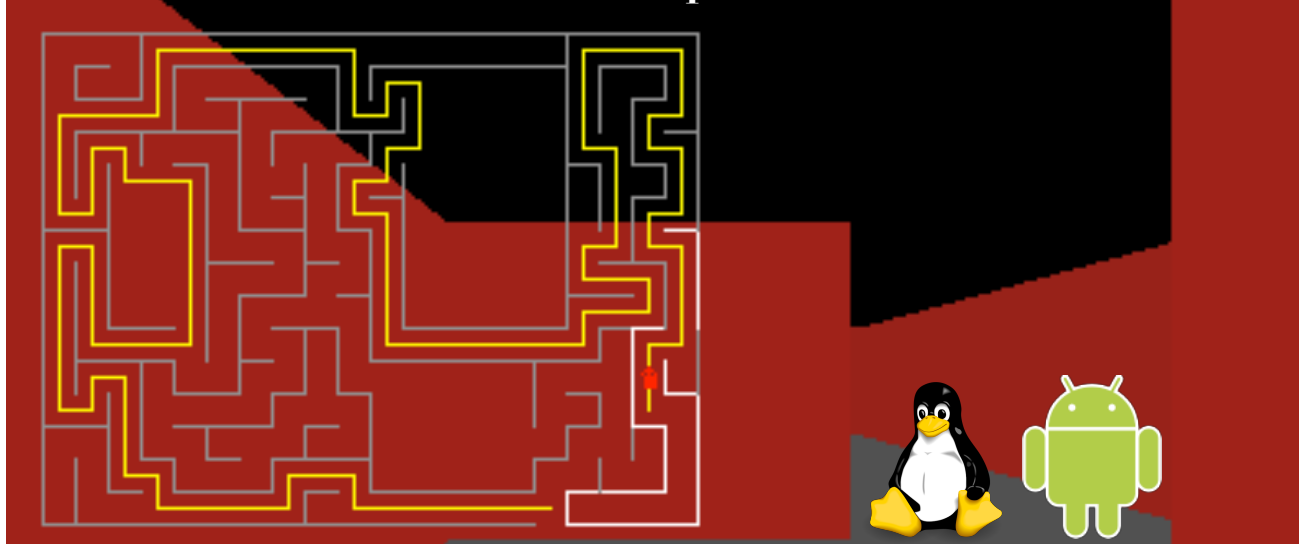


# CSCI 301 Software Development



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."

**Online course:** The pedagogical format of this course is an online course that is fully asynchronous (no meetings where all students and instructor necessarily meet at the same time/location). All learning materials are available online. The course is self-paced within a coarse-grained overall weekly schedule for assessments and project due dates. A Piazza forum is offered as an online communication platform for questions & answers and discussions among students. The instructor participates in the Piazza forum and offers virtual office hours for meetings via Skype or alike (as well as office hours for physical meetings if students request it). All tests and final exam are online, there is no need to be at W&M.

**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, Subversion for versioning) will be introduced and used in homework and project assignments.

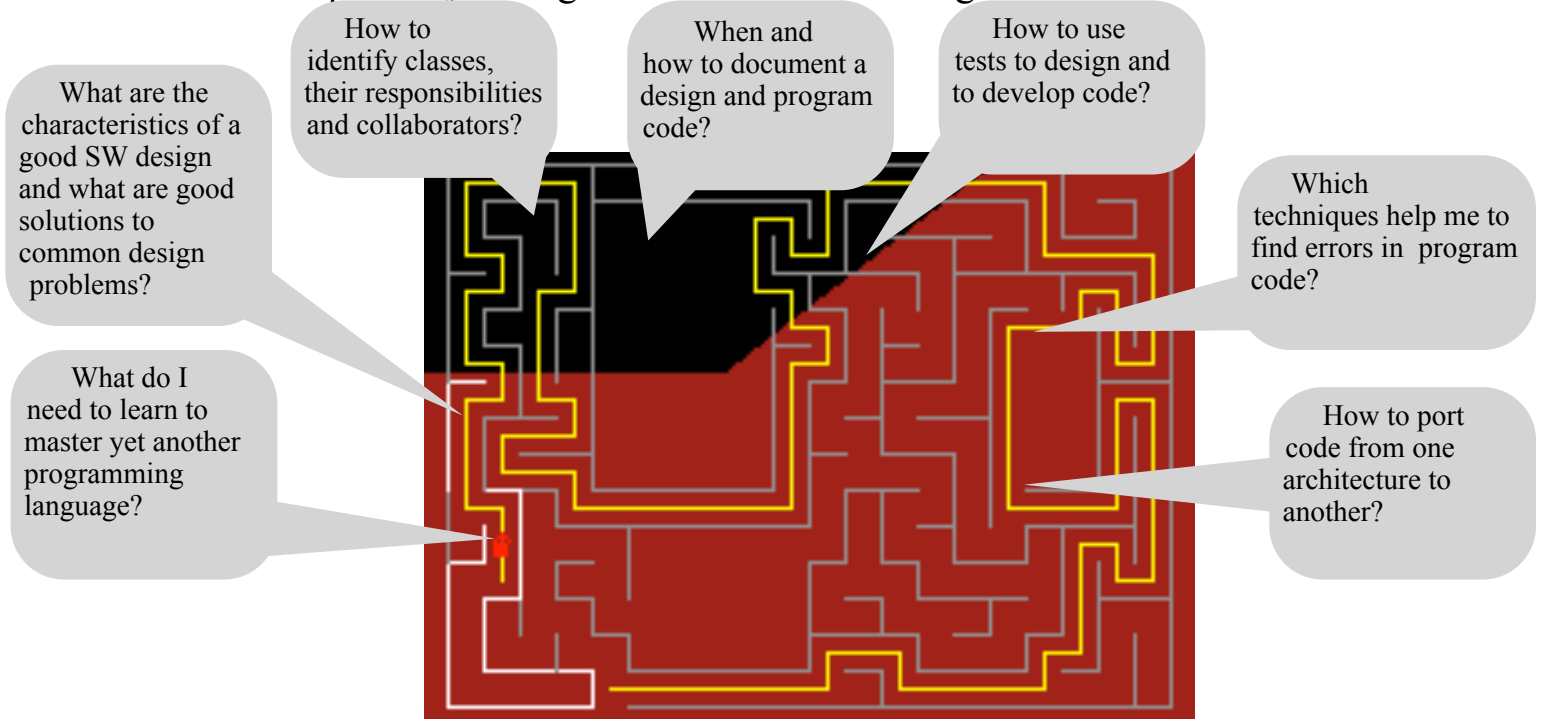
The second half of the course will cover software development for mobile applications, in particular object-oriented programming in Java for App development on [Android](#). The software development will use the Android emulator that interacts with the Android Studio IDE such that no additional hardware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

# CSCI 301 Software Development

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.

## 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: +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 five projects assignments that 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, but source code remains private and is not shared.

Tests & Final Exam (20%): There will be weekly tests (4 tests, 2.5% each, 10% in total) and a final exam (10%) posted on blackboard.

## **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. Deadlines will be set well in advance and announced with project assignments. The current plan is to have weekly project assignments and deadlines.

## **Students who need accommodation**

Please communicate your accommodation needs with me.

## **Information Dissemination**

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