



Mark Austin,  
Department of Civil Engineering,  
University of Maryland, College Park.

### [Notes from Class](#)

Meet the Class: [ [2012](#) ] [ [2013](#) ] [ [2016](#) ]  
[ [2017](#) ] [ [2018](#) ] [ [2019](#) ]

Projects: [ [2012](#) ] [ [2013](#) ] [ [2016](#) ] [ [2017](#) ]  
[ [2018](#) ] [ [2019](#) ]

## GOALS

This course will be a hands-on introduction to engineering software development for the model-based design and operational management of modern civil systems. Students will learn how to model the structure and behavior of civil systems, and then develop object-oriented software solutions for specific civil systems applications. Motivating case studies will be drawn from road, rail, and utility networks, networked building services, and spatial modeling for buildings and urban areas.

[Click here](#) to see a sample of screenshots from recent class projects.

---

## COURSE CONTENTS - SPRING SEMESTER 2019

The topics will be as follows:

### **Part 1: Data and Information Management for Modern Civil Systems (2 weeks)**

- **Modern Civil Systems**

- Topic: Features and required capabilities.

- Topic: Role of sensing, communications, control, and computation.

- Topic: Pathway from sensing to data to information, knowledge and decision making to action.

- Topic: Cyber-physical systems.

- Topic: Challenges in development and operational management.

- **Visual Modeling with UML**

- Topic: History and Goals of UML.

- Topic: Representation of concepts, relationships, constraints, rules and operations.

- Topic: Modeling abstractions for system structure and system behavior.

- **Motivating Case Studies**

- Architecture and behavior modeling for the Washington D.C. Metro System.

- Object and topological relationships in the GIS domain.

- Applications of the Open Street Map Markup Language (OSM).

## Part 2: Object-Oriented Software Development (3 weeks)

- **Languages for Engineering Software Development**
  - Topic: Evolution of computer languages over the past 20 years?
  - Topic: Features of low- and high-level languages
  - Topic: Features of scripting languages versus compiled languages
- **Getting Started with Java**
  - Topic: Writing and Compiling a Simple Java Program,
  - Topic: Basic programming (data types, expressions, assignments, branching constructs, loops),
  - Topic: Public and private methods and data.
  - Topic: Single- and multi-dimensional arrays,
  - Topic: Software Productivity Tools: Ant
  - Topic: Integrated Development Environments: Eclipse, Netbeans, Xcode 5/6.
- **Object- and Component-Based Modeling**
  - Topic: Classes, objects, association relationship.
  - Topic: Inheritance and abstract classes.
  - Topic: Guidelines for class and package design.
- **Introduction to Data Structures and Algorithms**
  - Topic: Arrays, arraylists, hash tables, and trees.
  - Topic: Working with the Java Collections Framework.

## Part 3: Modeling System Structure and System Behavior (2 weeks)

- **Modeling System Structure**
  - Topic: Association relationships (e.g., many-to-one, many-to-many),
  - Topic: Graph and network data structures and algorithms.
  - Topic: Two- and three-dimensional spatial representations.
- **Modeling System Behavior**
  - Topic: Finite state machines and statecharts.
- **Case Study**
  - Topic: Structure and Behavior Modeling for the Washington D.C. Metro System

## Part 4: Design Patterns and Graphical User Interfaces (GUIs) (2 weeks)

- **Introduction**
  - Topic: Design patterns in architecture and city planning.
  - Topic: Definition and Importance
- **Structural Design Patterns**
  - Topic: Composite, adapter, and bridge design patterns.
- **Behavioral Design Patterns**
  - Topic: Observer and mediator design patterns.
- **System Design Patterns**
  - Topic: Model-view-controller and router design patterns.
- **GUI Development with JavaFX**
  - Topic: Organization of JavaFX components.
  - Topic: Layout managers, event listeners.
  - Topic: Working with Standard Geometric Shapes.
- **Case Study**
  - Topic: Visualization of networks and graphs.

## Part 5: Advanced Topics (2 weeks)

Depending on student interests and project needs, topics will be selected from:

- **Integration of Software Systems with Python**

Topic: Features and benefits of Python.

Topic: Writing and running a simple Python programs.

Topic: Working with Jython.

- **More Graphical User Interface Development**

Topic: Working with JavaFX.

Topic: Working with free plotting software (e.g., matplotlib).

- **Concurrent Behavior Modeling with Threads**

Topic: Working with the Runnable Interface.

Topic: Lifecycle of a threaded process.

Topic: Threads and observers.

Topic: Opportunities and applications.

- **XML binding for Java (JAXB)**

Topic: Java to XML conversion.

Topic: XML to Java conversion.

- **Working with Semantic Web Technologies**

Topic: The Resource Description Framework (RDF).

Topic: The Web Ontology Language (OWL).

Topic: Semantic Modeling with Jena and Jena Rules.

Students will complete individual homework assignments, and work in small teams on a Civil Systems software development project.

## COURSE PREREQUISITES

- Graduate level status in engineering.
- An introductory knowledge of computers and hands-on computer programming.
- A good knowledge of engineering mathematics (e.g., calculus, linear algebra, differential equations).

---

## TIME AND LOCATION OF CLASS/OFFICE HOURS

- **Class.** EGR 3114, Tuesday and Thursday, 5 - 6.15 pm.
- **Office Hours.** By appointment. For a quick response, send me an e-mail.

---

## COURSE ASSESSMENT AND EXAMINATION SCHEDULE

The course will be assessment will be as follows:

- Homework (20%).
- One midterm exam (25%).  
The exam will be open book and open notes.
- Final exam (25%): May XX, 5-7pm in our regular classroom.  
The exam will be open book and open notes and will be similar in style to the midterm exam.  
Study hints have been posted to the notes-from-class web page.
- End-of-semester project/report involving the use of graphs/networks (30%).

**Note.** Students may drop the midterm exam if they do better on the final exam (i.e., it can count for up to 50% of the final grade).

---

## TEXTBOOKS AND SUPPORT MATERIAL

- **SOFTWARE DEVELOPMENT WITH JAVA**  
Liang Y.D., Introduction to Java Programming (Comprehensive Version), 8th Edition, Prentice-Hall, 2011.
  - Horstmann C., [Object-Oriented Design and Patterns](#) , 2nd Edition, John-Wiley and Sons, 2003.
  - James Gosling, et al., [The Java Language Specification: Java SE 7 Edition](#) , 2013.
  - James Gosling, et al., [The Java Language Specification: Third Edition](#) , 2005.
  - **SOFTWARE DEVELOPMENT WITH PYTHON**  
Liang Y.D., Introduction to Python Programming, Prentice-Hall, 2012.
  - Beazley D.M., Python: Essential Reference (Fourth Edition), Addison Wesley, 2009.
  - **CLASS LECTURE NOTES**
  - The class lecture notes are now available from the Engineering Copy Center.  
Cost: \$55 (Cash or Check made out to "University of Maryland").
  - **SUPPORT MATERIAL**  
Collections of research papers and presentations on the various topics covered in class will be distributed in class.  
In recent semesters I have simply passed a CD/memory stick around the classroom.
- 

## BACKGROUND TEXTS

### Civil Systems Modeling

- Eastman C., Teicholz P., Sacks R., and Liston K., **BIM Handbook** , John Wiley and Sons, 2008.
- Butler J.A., **Designing GeoDatabases for Transportation** , ESRI Press, 2008.
- Zeiler M., **Modeling our World: The ESRI Guide to Geodatabase Design** , ESRI Press, 1999.

### Design Patterns

- Stelling S. and Maassen O., **Applied Java Patterns** , The SUN Microsystems Press/Prentice-Hall, 2002.
- 

## CIVIL SYSTEMS ON THE WEB

- Open-Air Computers: [Cities are turning into Vast Data Factories](#)  
Economist, October 27, 2012.
  - [High-Tech Sensors Help Old Port City Leap Into Smart Future](#)  
NPR, June 4, 2013.
  - [SIEMENS](#) : Building Sustainable Cities
  - [Living PLaNIT Technologies for Sustainable Cities](#)
- 

## JAVA AND PYTHON SOURCE CODE

- [Getting Started with Java](#)
- Working with [Java Swing and Java 2D](#)
- Working with [Java Collections](#)

In class, Java/Python code will be distributed for:

- Working with software design patterns,
- Working with graph and networks,
- Working with semantic web (i.e., rule-based reasoning with ontologies and rdf graphs),

- Working with two-dimensional GIS (i.e., Java Topology Suite),
  - Working with solid models,
  - Working with symbolic expressions, and
  - Working with physical units.
- 

## SOFTWARE TOOLS AND LIBRARIES

- [Java 2D: An Introduction and Tutorial](#) .
- [Working with Java3D and Scenegraphs \(from the UK\)](#)
- [Java programming source code examples \(www.java2s.com\)](#)
- [Python programming source code examples \(www.java2s.com\)](#)
- [Wicked Cool Java \(www.wickedcooljava.com\)](#)
- [Downloads for Wicked Cool Java](#)
- [Java Topology Suite](#) (Robust algorithms for two-dimensional GIS)
- [Google SketchUp](#) .
- [Numeric and Scientific Programming in Python](#) .
- [Python 3D Software Collection](#) .
- [Scientific Tools for Python \(www.scipy.org\)](#) .
- [GME Tutorials at Vanderbilt](#) (see short tutorials for construction of networked blocks).
- [Libraries for Constraint Programming](#)
- [Graph Databases](#) listed at Wikipedia.  
Take a look at [Neo4J](#), a graph database from Sweden that scales to billions of nodes, relations and properties, and works with both Java and Python.

Last Modified: February 19, 2018.

Copyright © 2012-2017, Department of Civil and Environmental Engineering, University of Maryland