Engineering 9874
Software Design and Specification

Introduction

Dennis Peters
dpeters@mun.ca

Winter 2008

Administrivia

Evaluation:	**Assignments** (2) :	10%
	Project (2 parts):	30%
	Mid-term test :	10%
	Final Exam :	50%

Lectures: Monday, Wednesday, Friday 0900-0950 EN-4033

Office hour: Wednesday 1500–1600, Friday 1400-1500, or by appointment (or not).

Web page: http://www.engr.mun.ca/˜dpeters/9874/

Teaching Assistant: T.B.D.

Objectives

Some classes of software development:

- in the small — working alone to develop a small program with only a few functions or classes (say ≤ 1 KLOC).
- in the large — working as a member of a team (usually ≥ 4 people) to develop a substantial software system (say ≥ 20 KLOC).

This course seeks to bridge between these two. It addresses the issues of system structure and class design.

Objectives (cont'd)

From this course, students should gain the following:

- Background in the key concepts of professional object-oriented software design and development.
- Practical experience in object-oriented software design and development.
- An understanding of the Unified Modelling Language.
- Practice in using UML to specify, design and implement a significant project.

Outline

- Modelling and Design using UML
  - Class Diagrams
  - Interaction Diagrams
  - Use-cases and requirements
- Class Relationships
  - Design Patterns
  - Frameworks
- Specification
  - Object Constraint Language
  - Statecharts
- Java classes and packages
- Verification & validation

Key Dates and Evaluation

Date	Item	Marks
Fri. Jan. 18	Assignment 1	5%
Fri. Feb. 1	Assignment 2	5%
Fri. Feb. 15	Mid-term test	10%
Fri. Mar. 7	Project part 1	15%
Fri. Mar. 28	Project part 2	15%
T.B.D. (Apr. 9–18)	Final exam	50%

Main Books

[1] Bernd Bruegge and Allen H. Dutoit.
Object-Oriented Software Engineering: Using UML, Patterns, and Java.
Pearson Prentice Hall, second edition, 2004.

[2] Robert C. Martin.
Agile Software Development: Principles, Patterns, and Practices.
Prentice-Hall, 2003.

[3] Jos Warmer and Anneke Kleppe.
Object Constraint Language: Getting Your Models Ready for MDA.
Addison-Wesley, second edition, 2003.

Other References

[1] David Flanagan.
Java in a Nutshell.
O'Reilly & Associates, fifth edition, 2005.

[2] Martin Fowler.
UML Distilled: A Brief guide to the Standard Object Modeling Language.
Addison-Wesley, third edition, 2004.

[3] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides.
Design Patterns: Elements of Reusable Object-Oriented Software.
Addison-Wesley, 1994.

Software

- Java Software Development Kit (at least Sun version 1.4.2) — available from java.sun.com. You may also want an IDE such as Eclipse (www.eclipse.org) or JBuilder and you might find Ant (ant.apache.org) helpful.
- JUnit (www.junit.org)
- UML tool (e.g., Rational Rose, Poseidon (www.gentleware.com), eUML2 (www.soyatec.com/euml2))