

# **OMSE 533 – Software Design Techniques**

## **Winter 2005**

### **Preliminary Syllabus**

This is a **preliminary** syllabus for OMSE 533; a more detailed syllabus will be distributed when class starts. I am in the process of updating the course material so you can expect that it will change in some details as class progresses.

#### **Synopsis**

While design methods and notations change relatively rapidly, the fundamental principles of software engineering and software design do not. The focus of this course is on understanding, and learning to apply, fundamental design principles. The student will learn to apply that understanding to evaluate design methods and to apply a given method or notation to achieve his or her design goals. The course will treat object-oriented design and the UML notation in some depth to give hands on experience in the application of fundamental principles in a particular design style and notation.

#### **Prerequisites:**

OMSE 500 or permission of instructor

#### **Resources:**

For this course (and all OMSE courses, for that matter) it is assumed that you have access to email, the Internet, and a computer running Microsoft Word ®. You will also need an Odin account to access Portland State's WebCT system.

#### **Objectives:**

On completion this course, students shall have the skills to:

- Describe the role of analysis and design in software engineering.
- Understand and be able to apply fundamental design principles including abstraction, information hiding, and separation of concerns.
- Apply design principles and heuristics to evaluate a software design
- Evaluate the “fit to purpose” of a design method relative to a given set of design objectives.
- Develop an OO design in UML that is consistent with design goals
- Understand the nature of design patterns.

#### **Required Texts:**

[Budgen] *Software Design (2nd Edition)*, David Budgen, Addison-Wesley Pub Co; ISBN: 0201722194, 2 edition (May 15, 2003).

[Parnas] *Software Fundamentals: Collected Papers by David L. Parnas*, David M. Weiss, Daniel M. Hoffman (Editors), Addison-Wesley Pub Co; ISBN: 0201703696; 1st edition (April 9, 2001)

#### *Optional*

For a concise summary of UML if you do not already have a UML book or previous experience.

[Fowler] *UML Distilled, Third Edition*; Martin Fowler; Addison Wesley, 3rd edition (September 19, 2003); ISBN 0321193687.

## Readings:

Readings are listed **in the week they are due – this means they should be read before the class period.** Unfortunately, this means that some students will need to double up the reading between weeks 1 and 2 but the once-a-week schedule makes this unavoidable. However, these early readings are not technically dense and can be gone through fairly quickly.

In addition to readings from the texts, we will occasionally read seminal articles from the literature. Copies of these articles will always be provided in class or electronically, usually the week before they are due.

## Preliminary Schedule

Week	Topic	Reading
1	Introduction to Software Design <ul style="list-style-type: none"><li>• What is “software design”: an introduction</li><li>• What makes software design difficult</li><li>• The role of Software Engineering</li><li>• Need for a “rational process”</li></ul>	- Brooks, “No silver Bullet,” (review) - Faulk, “Software Requirements” (review) - [Parnas] Ch 18, “A Rational Design Process” - [Budgen] Chapter 1
2	Software Design Principles <ul style="list-style-type: none"><li>• Principles vs. heuristics</li><li>• General Design Principles: Separation of concerns and Abstraction</li><li>• Specific Software Principles: Modularity, Information hiding</li></ul>	- [Parnas] Ch 9, “Concept of Transparency” - [Parnas] Ch 11, “Abstract types” - [Parnas] Ch 13, “Some SE Principles” - [Budgen] Chapter 2
3	Modular Decomposition <ul style="list-style-type: none"><li>• Policy vs. Mechanism</li><li>• Modular decomposition</li><li>• Maintainability</li></ul>	[Parnas] Ch 7, “On the criteria...” [Parnas] Ch 16, “The Modular Structure...” [Budgen] Ch 4
4	Decomposing to Design Goals <ul style="list-style-type: none"><li>• Fun with modules (applied modularization)</li><li>• What is “good” design?</li></ul>	[Parnas] Ch 10, “Program Families” [Parnas] Ch 14, “Extension and contraction” [Budgen] Ch 5
5	Design Methods <ul style="list-style-type: none"><li>• Characterizing module interfaces</li><li>• Design Methods: Functional Decomposition, SA/SD</li></ul>	Structured Design paper [Budgen] Ch 7 thru 7.25, 8, 9
6.	Object-Oriented Design I <ul style="list-style-type: none"><li>• OOD concepts</li><li>• Analysis and Use Cases</li><li>• OOD with UML: Static Structures</li></ul>	[Budgen] rest of Ch 7
7.	Object Oriented Design II <ul style="list-style-type: none"><li>• Dynamic structures</li><li>• Detailed Design</li><li>• Applying design principles in OO</li></ul>	TBD

<b>8.</b>	Advanced Design Concepts <ul style="list-style-type: none"> <li>• Evaluating Design</li> <li>• Patterns and frameworks</li> </ul>	[Budgen] Ch 10 [Parnas] Ch 12, "Undesired Events" [Parnas] Ch 17, "Active design reviews"
<b>9.</b>	Addressing strategic design goals <ul style="list-style-type: none"> <li>• Software Product-Lines</li> </ul>	Handout on Product-Lines
<b>10.</b>	<ul style="list-style-type: none"> <li>• Ethics and responsibilities</li> <li>• Course Review</li> </ul>	[Parnas] Ch 28, "Professional responsibilities"
<b>11.</b>	Final	