CSG 260

September 9, 2003

Advanced Software Development — Syllabus Fall 2003
Professor Karl. J. Lieberherr
Course home page:
http://www.ccs.neu.edu/home/lieber/courses/csg260/f03/f03.html

The syllabus is subject to change based on class reaction and projects selected by the students.

The Generative Programming book (GP book) is required (see course home page).

The AP book is the recommended text book: "Adaptive Object-Oriented Software: The Demeter Method" which you can browse at:

`http://www.ccs.neu.edu/research/demeter/biblio/dem-book.html.`

You don't have to buy the book.

This course provides state-of-the-art techniques and concepts for advanced software development.

In this incarnation of the course "advanced" means "aspect-oriented".

```
aosd.net
and
http://www.ccs.neu.edu/home/lieber/AOP.html.
```

You learn the foundations of aspect-oriented software development (AOSD) in a practical context. You learn about generative programming in general and aspect-oriented and adaptive programming in particular.

In the first few weeks you learn AOSD (using AspectJ and DJ) and the Demeter method which you apply in the remaining weeks to your project. The project will be done in groups.

Designing and programming will be done in a structure-shy, grammar-based, object-oriented style. All assignments and the project will be written directly or indirectly in AspectJ, a small extension to Java.

This course does not have Java or AspectJ as a prerequisite, however you are expected to learn a small subset of Java and AspectJ from resources on the web or from a good AspectJ book of your choice.

The WWW directory for the course is:

http://www.ccs.neu.edu/home/lieber/courses/csg260/f03/

This file is in:

admin/

1. Week: Introduction to Generative Programming, Aspect-Oriented and Adaptive Programming. The basics of AspectJ and adaptive software development. Static and dynamic join point model. The bus simulation example.

   - Pattern: Structure-shy Traversal.
     http://www.ccs.neu.edu/research/demeter/adaptive-patterns/pattern-lang-conv file: Structure-Shy-Traversal.html
   - Pattern: Selective Visitor.
     http://www.ccs.neu.edu/research/demeter/adaptive-patterns/pattern-lang-conv file: Selective-Visitor.html
   - Pattern: Class Graph (Class dictionaries and Unified Modeling Notation.)
     http://www.ccs.neu.edu/research/demeter/adaptive-patterns/pattern-lang-conv file: Class-Graph.html
   - Overview of pattern language, see: http://www.ccs.neu.edu/research/demeter/adaptive-patterns/pattern-lang-conv file: pattern-language-for-AP.html

   AP book chapters: selections from 1-4. GP book: Chapter 1, Chapter 2: Analysis and Design Methods and Techniques

2. Week:

- Pattern: Structure-Shy Object
  http://www.ccs.neu.edu/research/demeter/adaptive-patterns/pattern-lang-conv file: Structure-Shy-Object.html
- Pattern: Growth Plan
  http://www.ccs.neu.edu/research/demeter/adaptive-patterns/pattern-lang-conv file: Growth-Plan.html

   Writing simple adaptive programs using traversals and visitors. Introduction to DJ and connection to AspectJ. Class dictionary design.

   AP book chapters: selections from 5-8. GP book: Chapter 3: Domain Engineering and Object-Oriented Analysis and Design and Chapter 4: Feature Modeling

3. Week:

   Strategy graphs. Decoupling classes: Law of Demeter. Expressing Law of Demeter in AspectJ. Relationship to adaptive software. DJ continued. The class dictionary notation (graphical and textual). Design rule checking of class dictionaries. Class dictionaries as customizers for adaptive programs.

   AP book chapters: selections from 8-11. GP book: Chapter 5: The Process of Generative Programming and Chapter 6: Generic Programming

4. Week:

   Visitor pattern. Adaptive Programming. Traversal strategies overview. More complex uses of visitors: multiple visitors and nested visitors. Improving the reusability of software designs. Parameterized class definitions.

   AP book chapters: selections from 10-12. GP book: Chapter 8: Aspect-Oriented Programming and Chapter 9: Generators

5. Week: Traversal strategies in detail. The Demeter Method. Design notations for behavior. Growth Plan pattern. Developing a growth plan for implementation and testing. Design with maintenance in mind.

   AP book chapters: 13. GP book: Chapter 13: Bank Account

6. Week: XAspects. The coordination aspect. Exception handling design.

   Introduction to Eclipse. Testing of object-oriented software.

7. Week: Midterm. Advanced features of AspectJ.
8. Week: Aspect-Oriented Design. Eclipse continued.
9. Week: Aspectual Collaborations. Personalities. Further design patterns for adaptive object-oriented software development.

   http://www.ccs.neu.edu/research/demeter/adaptive-patterns/
10. Week: A comparison of aspect-oriented software development methodologies.
11. Week: Caesar compared to AspectJ: How to map use cases to aspects.
12. Week: Aspectual domain-specific languages, Genvoca
13. Week: Domain Engineering, Model-Driven Architecture
14. Week: Feature Modeling, Model-Driven Architecture
15. Exam week: Final.