CS 8803F: Object-Oriented Systems and Languages
Fall 2001 Syllabus

Meeting Times: T-Th, 4:35-5:55

Instructor: Yannis Smaragdakis, CCB 215, 404-385 1491
yannis@cc.gatech.edu
Office Hours: by appointment (walk-ins welcome :-)

Prerequisites: This is an advanced topics course. Prerequisites are not enforced, but you should have knowledge of programming languages equivalent to an undergraduate PL survey course and fluency in at least one OO language.

Evaluation: Grades will be based on a programming project, one or two in-class presentations, and (possibly) exams, depending on the size of the class and the distribution of students (i.e., the number of masters students, the number of non-CS students, etc.).

Course Outline:

The course will cover an extensive sample of work from the Object-Oriented Systems and Languages literature—mostly the OOPSLA and ECOOP conferences. Depending on student interests, the course emphasis may switch. Areas to be covered include:

Design Patterns—a critical view and analysis (not tutorial)

Aspect-Oriented Programming, Subject-oriented Programming, Adaptive Programming

OO Type Systems: parameterization mechanisms for Java, virtual types, module systems

Language Extensibility: meta-object protocols, reflection

Implementation issues: locality of OO programs, garbage collection for Java

Reading Material:

There is no textbook for this course. Papers from the literature will be used. The reading list, below, offers a sampling of selected papers and books. The list is not entirely complete (does not include very recent papers and some of the more systems-oriented material) but a large number of the papers discussed during the semester will come from this list.

Reading List:

O. Agesen, S. Freund, and J. Mitchell, “ Adding Type Parameterization to the Java Language”, *OOPSLA 1997*, 49-65.

D. Batory, V. Singhal, M. Sirkin, and J. Thomas, “Scalable Software Libraries”, *ACM SIGSOFT* 1993.

K. Beck and W. Cunningham, “ A Laboratory for Teaching Object-Oriented Thinking”, *OOPSLA 1989*, 1-6.

T.J. Biggerstaff, “The Library Scaling Problem and the Limits of Concrete Component Reuse”, *3rd Int. Conf. on Softw. Reuse (ICSR ‘94).*

G. Bracha and W. Cook, “Mixin-Based Inheritance”, *ECOOP/OOPSLA 1990*, 303-311.

G. Bracha, M. Odersky, D. Stoutamire and P. Wadler, “Making the future safe for the past: Adding Genericity to the Java Programming Language”, *OOPSLA 1998.*

K.B. Bruce, M. Odersky, and P. Wadler, “A Statically Safe Alternative to Virtual Types”, *ECOOP 1998*.

L. Cardelli and P. Wegner, On Understanding Types, Data Abstraction, and Polymorphism, *Computing Surveys*, 17(4): Dec 1985, 471-522.

S. Chiba, “Open C++ Programmer's Guide for Version 2”, SPL-96-024, Xerox PARC, 1996.

K. Czarnecki and U. Eisenecker. *Generative Programming: Methods, Techniques, and Applications*. Addison-Wesley, 2000.

K. Czarnecki and U. Eisenecker, "Synthesizing Objects", *ECOOP 1999*, 18-42.

M.A. Ellis and B. Stroustrup, *The Annotated C++ Reference Manual*, Addison-Wesley, 1990.

R.B. Findler and M. Flatt, “Modular Object-Oriented Programming with Units and Mixins”, *Int. Conf. on Functional Programming*, 1998.

M. Flatt, S. Krishnamurthi, M. Felleisen, “Classes and Mixins”. ACM *Symposium on Principles of Programming Languages*, 1998 (PoPL 98).

I.R. Forman, S. Danforth, and H. Madduri, “Composition of Before/After Metaclasses in SOM”, *OOPSLA 1994*.

E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

J. Goguen, “Reusing and interconnecting software components”, *IEEE Computer*, February 1986, 16-28.

James Gosling, Bill Joy, Guy L. Steele, *The Java Language Specification*, Addison-Wesley, Reading, Massachusetts, 1996.

W. Harrison and H. Ossher, “Subject-Oriented Programming (A Critique of Pure Objects)”. *OOPSLA 1993*, 411-428.

R. Helm, I. Holland, and D. Gangopadhyay, “Contracts: Specifying Behavioral Compositions in Object-Oriented Systems”. *OOPSLA 1990*, 169-180.

I. Holland, “Specifying Reusable Components Using Contracts”, *ECOOP 1992*, 287-308.

R. Johnson and B. Foote, “Designing Reusable Classes”, *Journal of Object-Oriented Programming*, 1(2): June/July 1988, 22-35.

R. Keller, U. Hoelzle, “Binary Component Adaptation”, *ECOOP 1998*.

G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J. Loingtier, and J. Irwin, “Aspect-Oriented Programming”, *ECOOP 1997*, 220-242.

G. Kiczales, J. des Rivieres, and D. G. Bobrow, *The Art of the Metaobject Protocol*, MIT Press, 1991.

K.J. Lieberherr, *Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns*, PWS Publishing Company, Boston, 1996.

O. L. Madsen and B. Møller-Pedersen, “Virtual classes: A powerful mechanism in object-oriented programming”, *OOPSLA 1989*, 397-406.

O. L. Madsen, B. Møller-Pedersen, and K. Nygaard, *Object-Oriented Programming in the BETA Programming Language*. Addison-Wesley, 1993.

M. Mezini, “Dynamic Object Evolution without Name Collisions”, *ECOOP 97*, 190-219.

M. Mezini and K. Lieberherr, “Adaptive Plug-and-Play Components for Evolutionary Software Development”, *OOPSLA 1998*.

D.A. Moon, “Object-Oriented Programming with Flavors”, *OOPSLA 1986*.

A. Myers, J. Bank and B. Liskov, “Parameterized Types for Java”, ACM *Symposium on Principles of Programming Languages*, 1997 (PoPL 97).

M. Odersky and P. Wadler, “Pizza into Java: Translating theory into practice”, *ACM Symposium on Principles of Programming Languages*, 1997 (PoPL 97).

H. Ossher and W. Harrison, “Combination of Inheritance Hierarchies”, *OOPSLA 1992*, 25-40.

H. Ossher, M. Kaplan, W. Harrison, A. Katz, and V. Kruskal, “Subject-Oriented Composition Rules”, *OOPSLA 1995*, 235-250.

C. Prehofer, “Feature-Oriented Programming: A Fresh Look at Objects”, *ECOOP 1997*, 419-443.

T. Reenskaug, E. Anderson, A. Berre, A. Hurlen, A. Landmark, O. Lehne, E. Nordhagen, E. Ness-Ulseth, G. Oftedal, A. Skaar, and P. Stenslet, “OORASS: Seamless Support for the Creation and Maintenance of Object-Oriented Systems”, *Journal of Object-Oriented Programming*, 5(6): October 1992, 27-41.

Y. Smaragdakis and D. Batory, “Implementing Reusable Object-Oriented Components”, *5th Int. Conf. on Softw. Reuse (ICSR ‘98)*, IEEE Computer Society Press, 1998.

Y. Smaragdakis and D. Batory, “Implementing Layered Designs with Mixin Layers”, *ECOOP 1998*.

L. Seiter, J. Palsberg, and K. Lieberherr, “Evolution of Object Behavior using Context Relations”, *ACM SIGSOFT* 1996.

Silicon Graphics Computer Systems Inc., *STL Programmer's Guide*. See: http://www.sgi.com/Technology/STL/ .

C. Simonyi, "The Death of Computer Languages, the Birth of Intentional Programming", *NATO Science Committee Conference,* 1995.

A. Stepanov and M. Lee, "The Standard Template Library", 1995. Incorporated in ANSI/ ISO Committee C++ Standard.

P. Steyaert, W. Codenie, T. D'Hondt, K. De Hondt, C. Lucas, and M. Van Limberghen, "Nested Mixin-Methods in Agora", *ECOOP 1993*, 197-219.

B. Stroustrup, *The C++ Programming Language, 3rd Ed.*, Addison-Wesley, 1997.

K. Thorup, "Genericity in Java with Virtual Types", *ECOOP 1997*, 444-471.

M. VanHilst and D. Notkin, "Using C++ Templates to Implement Role-Based Designs", *JSSST International Symposium on Object Technologies for Advanced Software*, Springer-Verlag, 1996, 22-37.

M. VanHilst and D. Notkin, "Using Role Components to Implement Collaboration-Based Designs", *OOPSLA 1996*.