CS 6304: Topics in Object-Oriented Systems and Languages  
Fall 2006 Syllabus

Meeting Times: Tuesday and Thursday, 3:30-4:45pm

Meeting Place: Norris Hall 306

Instructor: Dr. Eli Tilevich, 213 Knowledge Works II, 540-231-3475
tilevich@cs.vt.edu
Office Hours: by appointment (walk-ins welcome :-)

Description: Object-oriented (OO) systems has been one of the most dynamic research areas in recent years. Beyond encapsulation, inheritance, and polymorphism, research in OO systems is a confluence of various topics in systems, programming languages, compilers, and software engineering. Specifically, research in OO systems has engendered several exciting recent developments in areas including: programming languages (e.g., genericity, reflection, meta-programming, bytecode engineering, virtual dispatch, garbage collection, just-in-time-compilation), middleware (e.g., distributed-object systems), concurrency (e.g., Java memory model), and many others. It has even spawned an entirely new research area: aspect-oriented programming (AOP). Knowledge in OO systems is essential for anyone involved in development of next generation technologies.

This course will provide students with a background in OO research by covering both standard research literature and providing hands-on experience with specific technologies. In addition, the course will introduce students to research opportunities in current state-of-the-art OO systems. Additional topics covered will be influenced by the research interests of the class.

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: a term paper or a project (60%)
hand-on exercises (20%)
a research paper presentation (10%)
class participation (10%)

Other Resources:
Listserv: CS6304_97199@listserv.vt.edu
Web Page: http://courses.cs.vt.edu/~cs6304/fall.06/

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)

Component-based designs: layered design, mixin, mixin layers

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: efficient dynamic dispatch, multiple inheritance and object layout, garbage collection for Java

Software tools: OO middleware, bytecode engineering, code generation, OO distributed systems

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 complete (does not include very recent papers) but it is meant to give you a taste of the material we will study.

Reading List:


John K. Bennett, “The Design and Implementation of Distributed Smalltalk,” in *OOPSLA 1987*.

T.J. Biggerstaff, “The Library Scaling Problem and the Limits of Concrete Component Reuse”, *3rd Int. Conf. on Softw. Reuse (ICSR ’94)*.


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


