**CS 5984: Object-Oriented Systems and Languages**  
**Fall 2008 Syllabus**

**Meeting Times:** Monday, 3:30-4:45pm

**Meeting Place:** TBA

**Instructor:**  
Dr. Eli Tilevich, 213 Knowledge Works II, 540-231-3475  
tilevich@cs.vt.edu  
Office Hours: TBA

**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. Several novel programming paradigms such as aspect-oriented programming (AOP) stem from research in object-oriented technologies. Therefore, 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 technologies 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 determined by the individual interests of the class’s participants.

**Prerequisites:** Because this is a graduate course, prerequisites are not strictly enforced. However, you should have knowledge of programming languages equivalent to an undergraduate PL survey course and fluency in at least one OO language.

**Evaluation:**  
- hands-on exercises (20%)  
- a midterm exam (20%)  
- a research paper presentation (10%)  
- a term paper or project (40%)  
- class participation (10%)

**Other Resources:**  
**Listserv:** TBA  
**Web Page:** TBA

**Course Outline:**  
The course will cover an extensive sample of work from the Object-Oriented Systems and Languages literature such as can be found in the proceeding of the OOPSLA and ECOOP conferences. Specific 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

OO Distributed Systems: OO middleware, distributed object systems

Software tools: code generation, code transformation, bytecode engineering

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:


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.


