

## CSC 520 – Advanced Object Oriented Programming, Fall 2010 Syllabus

Grim 307, Th 6-8:50 PM

Dr. Dale E. Parson, Old Main 260, parson@kutztown.edu, 484-646-4296

Office Hours: M 11-12, Tu 4:15-5:45, W 11-12, Th 4:15-5:45, and by appointment

<http://faculty.kutztown.edu/parson/>

This course introduces the concepts of object-oriented programming languages, object-oriented analysis and design, and design patterns, and demonstrates their use in the development of an object-oriented implementation of a major project. The Unified Modeling Language (UML) is used to develop the project's design and implementation. A current programming language is used throughout the course to illustrate major concepts and implement the project.

**We will be programming in Java, modeling in UML, rapid prototyping in Jython.**

*Object Oriented Software Construction*, 2nd Edition, Bertrand Meyer, 1997.

Optional text: *UML Distilled*, Third Edition, Martin Fowler, 2004.

**Grading** (on an A-90, B-80, C-70, D-60 scale)

Midterm 15% of grade

Non-cumulative final 15% of grade

Programs 70% divided equally among the project assignments

### Assignment grading criteria

5% Concise, accurate documentation of the operations of an abstract data type interface signature, preconditions, postconditions, invariants

10% Clean separation of specification of interface from concrete module implementations

5% Documentation of client modules (including tests) that use ADT modules

30% Correct implementation of central data structure or algorithm

20% Adherence to written specification of assignment, including coding standards

5% Willingness to ask questions and clarify any uncertainties about assignments

5% Concise, structured, and accurate documentation of implementation

10% Inclusion of test code, data & results showing coverage of common and corner cases

10% Successful execution against my private evaluation test cases

**The academic integrity policy is at <http://cs.kutztown.edu/pdfs/AcademicIntegrityPolicy.pdf>**

Your first reading assignment is to read the above policy statement.

You may openly discuss ideas, algorithms, pitfalls, and the use of programming tools.

You may not share code, test drivers or test data except within groups for group projects.

Group projects, when assigned, have documented partitioning of student responsibilities.

I will be using the Moss detection system at <http://theory.stanford.edu/~aiken/moss/>.

Class attendance is not mandatory, but I will be teaching fundamentally from experience, using the textbook and some slides as additional sources of information. You are responsible for all material covered in class, including technical information, coding standards and conventions, verbal specification of assignments, and your questions about topics that are not clear to you. Please, there should be no classroom conversations, cell phones, text messaging, eating, sleeping, obscenities, listening to music or other disruptions of the class. Thank you.

Week	Chapters	Lecture Topics
1	6-8, 11	Modules, packages, ADT, classes, objects, design by contract
2	12, 14-15	Exceptions, interface-implementation-mixin inheritance, tools
3		Partitioning a design space, design patterns , refactoring
4	22, 23	Finding the classes, use cases, CRC cards, Agile development
5		Frameworks, distributed frameworks, deployment
6	10, 16, 23	Antipatterns , generics, library construction
7	UML	Sequence, collaboration, statechart, activity diagrams
8		<b>Midterm exam (15%),</b> project work
9		Dynamic class loading, reflection, run-time interpretation
10		Dynamically typed O-O languages, rapid prototyping, Jython
11		Object graph serialization, persistence, O-O databases
12		Multithreading, active objects, delayed evaluation
13		Multithreading continued, Remote Method Invocation
14		Consolidation and review, project work
15		<b>Non-cumulative final exam, 15% of grade</b>