

# CompSci 552: Object-Oriented Programming

John Boyland

Fall 2009  
TR 2:00–3:15pm

Instructor        John Boyland  
Office            925 EMS, 229-6986  
Office Hours     T 3:30–4:30pm, R 11am–12pm, or by appt.  
Course Homepage <http://www.cs.uwm.edu/classes/cs552>

## 1 Introduction

CompSci 552 teaches the Java programming language for those who are already competent in C++ and also introduces students to refactoring, design patterns and concurrency. The course will use a freeware IDE (Eclipse) and a distributed filesystem (AFS).

## 2 Intended Course Outcomes

Students will be proficient in object-oriented programming in Java, more specifically, they will be

1. able to describe software architecture using simple UML;
2. able to design and implement programs with many classes;
3. able to understand and use design-by-contract concepts, including preconditions, postconditions and invariants;
4. able to use standard library classes;
5. able to write interactive programs using AWT/Swing;
6. able to apply refactoring appropriately;
7. able to understand and use object-oriented design patterns, including TEMPLATE, STRATEGY, PROTOTYPE, BUILDER, FACTORY, COMPOSITE, DECORATOR, COMMAND, ADAPTER;
8. able to understand and prevent concurrency errors, including race conditions and deadlock;
9. familiar with implementation of distributed systems, including push and pull techniques.

## 3 Requirements

Students must have received a ‘C’ or better in CompSci 351, and should also have basic knowledge of Java, for example, as provided in CompSci 431. In particular, this class assumes the following abilities:

- able to define classes with information hiding (abstract data types ADTs)
- able to use inheritance and overriding for object-oriented polymorphism
- able to understand and modify programs with tens of classes
- familiar with programming in Java

## 4 Texts

The required textbook for the course is

Xiaoping Jia. *Object-Oriented Software Development Using Java*. 2nd ed. Addison-Wesley. 2003.

## 5 Grading

The grade for the course will be computed from the following parts:

### 10% Quizzes

There will be quizzes on the required reading at the start of class each week. Each quiz is 1% of the course grade. The lowest quiz grades will be dropped.

### 30% Exams

There will be one midterm and one final exam, each worth 15% of the grade.

### 60% Homework

There will be a homework due every two weeks (7 in all), partly programming and partly text. Each homework is due at 2:00pm on Tuesday or Thursday (before lecture). Late homework will not be accepted. Each homework is worth 10% of the grade. The lowest homework grade will be dropped.

Exceptions to these rules can be made only in extraordinary circumstances. Advance notice of a need for an exception should be given whenever possible.

### 5.1 Criteria

Programs and written assignments will be graded for correctness, suitability, style, clarity and practicality. Although we will provide solutions to some assignments, there are usually a wide variety of correct answers to any particular assignment.

### 5.2 Academic Honesty

Unless an assignment explicitly says otherwise, all graded assignments must be *your own work* (your own words), but you may work with other people as long as you list their names prominently on the first page of the assignment, and/or in a comment at the top of the assignment. For this course, verbal communication and collaboration using non-code text or hand-written code is permitted, as long as it is properly documented. Documentation must also be made for help from anyone not in the course, such as a tutor, friend, or relative, and for information off the Web.

You may not make copies for other people of assignments through email, disks, scanning or any other automatic copying technique, except where specifically permitted in an assignment. Whether or not you have permission of the other, submitting someone else's work as your own is plagiarism, a serious instance of academic misconduct. Everyone is responsible for learning the material themselves. Some of the assignments may be graded in person, especially in cases where the individual contribution to the assignment is not clear. If you are graded in person, you will be expected to demonstrate that you have mastered techniques used in the material you submitted.

In this course, you will be running code written by course staff. None of these programs are "Trojan horses": none will ever attempt to perform covert tasks that invade privacy or destroy data. Similarly, you may not submit Trojan horses as regular assignments. Any such action will be dealt with severely.

### 5.3 Course Grades

At the end of the course, the numeric grade will be converted into a letter grade according to the following scale:

Letter Grade	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F
Minimum Score	92	90	87	82	80	77	72	70	67	62	60	0

There is no curve, but the instructor reserves the right to increase a grade if he believes it would not reflect the student's mastery of the material. A grade can only be decreased for academic misconduct.

## 6 The Eclipse IDE

You may use the Eclipse IDE for your Java programming. It is much easier than using emacs plus javac, but it is also resource intensive. Eclipse is installed in AFS in the class area, and it has been requested to be installed on the Windows build on the lab computers, but we recommend you install it on your home computer. In particular, you can make sure your home computer has OpenAFS installed.

Eclipse lets you decide where you want to put your "workspace" and where to put the "projects." The projects should be placed as subdirectories of the homework area. The workspace can be anywhere. (In particular you are permitted to use the homework directory itself as the workspace.)

## 7 Special Features of the Filesystem

All your work should be done in the AFS volume assigned to you:

```
/afs/cs.uwm.edu/users/classes/cs552/001/PantherID.
```

You may wish to make a symbolic link from a UNIX account to this directory. Before you can access this directory, you will need to obtain authenticate yourself to AFS using Kerberos. This can be done using

```
/usr/afsws/bin/klog PantherID
```

and type your Kerberos password. The `kpasswd.uwmcs` command in the same directory can be used to change your kerberos password. (The command `kpasswd` won't work. One version of this command simply will claim that you didn't type the old password correctly. The other will change your password but will make it impossible to 'klog'.)

The root directory of your volume has some special subdirectories:

**OldFiles** The directory includes a copy of your volume taken at 1am. If you accidentally delete a file that you need, please look for a backup here before contacting lab staff.

**Grades** This directory includes a file giving the grade for each assignment. The first line gives the numeric grade and the remainder of the file gives comments. The file `summary.txt` lists all the scores and computes the weighted percentage.

If you are told to do the assignment with another person, you can give them permission to write files in a subdirectory such as `homework4` by typing

```
/usr/afsws/bin/fs sa homework4 someoneelse write
```

## 8 Schedule

Date	Topic	Due Tuesday	Due Thursday	Reading
9/3	Introduction; UML			Ch. 1, 2
9/8	Java Basics			Ch. 3, 4
9/15	Java Classes and Inheritance	<b>Homework #1</b>		Ch. 5
9/22	Documentation and Invariants			Ch. 6
9/29	Generalization: TEMPLATE, STRATEGY	<b>Homework #2</b>		Ch. 7
10/6	Collections			Ch. 8.1–2
10/13	AWT and Swing; I/O Library	<b>Homework #3</b>		Ch. 8.3–4
10/20	<b>Midterm</b> ; Intro. to Refactoring			Ch. 9.1
10/27	Refactoring		<b>Homework #4</b>	Ch. 9.2–4
11/3	STATE, FACTORY, COMPOSITE			Ch. 9.5–7
11/10	PROTOTYPE, BUILDER, COMMAND		<b>Homework #5</b>	Ch. 10.2–3
11/17	ADAPTER			Ch. 10.4
11/24	Concurrent Programming			Ch. 11
12/1	Avoiding Concurrency Failure	<b>Homework #6</b>		lecture notes
12/8	Socket-Based Communication			Ch. 12.1
12/15	(No class)		<b>Homework #7</b>	
12/21	Final Exam (Monday 12:30pm)			

## 9 Reference

Online reference for Java:

- <http://java.sun.com/j2se/1.5.0/docs/api/>
- <http://java.sun.com/docs/books/tutorial/uiswing/index.html>
- <http://java.sun.com/docs/books/tutorial/>

## 10 Notes

- Eclipse is available at <http://www.eclipse.org>; free downloads are available for many operating systems.
- AFS is available at <http://www.openafs.org>; free downloads are available.
- If you will be needing accommodations in order to meet any of the requirements of this course, please contact us as soon as possible.