

CompSci 431: Programming Languages Concepts—Spring 2009

John Boyland

Instructor John Boyland
Office 925 EMS, 229-6986
Office Hours M 12:30–1:30pm, W 10:00–11:00am, R 1:00–2:00pm
Course Homepage <http://www.cs.uwm.edu/classes/cs431/>

1 Introduction

CompSci 431 introduces a variety of programming languages for those who already know at least one (we expect C++). The course briefly covers ML, Java and Prolog, but also gives a broad overview to the field of programming languages. In this section of the course, we will use a distributed filesystem (AFS).

2 Outcomes

1. Students will be able to understand and write context-free grammars using BNF/EBNF notations and syntax diagrams
2. Students will be able to understand core programming language concepts that include types, polymorphism, scopes, memory allocation of variables, memory management, parameter passing styles, and formal semantics.
3. Students will be able to understand and use the concepts of
 - (a) functional programming (pattern matching, type inference, parametric polymorphism, nested functions, higher-order functions, data and type constructors),
 - (b) object-oriented programming (inheritance, encapsulation, generics, subtyping polymorphism, dynamic dispatch, and exception handling)
 - (c) logical programming (resolution and unification).
4. Students will be able to write simple programs using the programming languages ML, Java, and Prolog.

3 Requirements

Students must have received a ‘C’ or better in CompSci 351. In particular, this class requires that student be able to use and implement

- standard data structures (dynamic arrays, linked lists, binary search trees) and
- standard abstract data types (lists, tables, graphs, stacks, queues).

4 Texts

The required textbook for the course is

Adam Brooks Webber. *Modern Programming Languages: A practical introduction*. Franklin, Beedle and Associates, Inc., 2003.

5 Grading

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

40% Exams

There will be one midterm and one final exam, each worth 20% of the grade.

60% Homework

There will be a homework every week (13 in all), partly programming and partly text. Each homework is due at the start of lecture. Late homework will not be accepted. Each homework is worth 5% 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 dishonesty.

6 Special Features of the Filesystem

All your work should be done in the AFS volume assigned to you. The root directory of the volume is accessible from your UNIX account using the `afs-home` link. The absolute pathname is

`/afs/cs.uwm.edu/users/classes/cs431/001/PantherID.`

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 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** 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
```

7 Schedule

Date	Topic	Reading
1/26	Introduction; Syntax	Ch. 1–2
2/2	Abstract Syntax; Language Systems	Ch. 3–4
2/9	Introduction to ML; Type	Ch. 5–6
2/16	ML Patterns and Nesting; Polymorphism	Ch. 7–8
2/23	Higher Order; Scope	Ch. 9–10
3/2	ML Datatypes; Activation Records	Ch. 11–12
3/9	Introduction to Java; Memory Management	Ch. 13–14
3/16	SPRING BREAK	
3/23	Midterm ; Java Classes	Ch. 1–14; Ch. 15
3/30	Object Orientation; Exceptions	Ch. 16–17
4/6	Parameters; Introduction to Prolog	Ch. 18–19
4/13	Procedural Prolog; Cost Models	Ch. 20–21
4/20	Arithmetic in Prolog	Ch. 22
4/27	Semantics	Ch. 23
5/4	History of Programming Languages	Ch. 24
5/11	10am: Final Examination	Ch. 1–24

8 Notes

- Resources (including copies of all slides) are available on the web.
- 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.