CS 354: Programming Languages

Instructor

Instructor:	Jim Buffenbarger
Electronic mail:	`buff@cs.BoiseState.edu`
Office:	MEC-302C 426–3567

Meetings

Lectures:	MoWe 12:00–1:15	ENGR-335
Office hours:	We 10:15–12:00	MEC-302C
	by appointment	MEC-302C

Our graduate assistants are Archana and Jared. You can find their schedules at:

`http://coen.boisestate.edu/cs/computer-science-tutoring-center-cstc`

Catalog Description

Principles of programming languages: design, syntax, semantics, information binding, strings, arithmetic, input/output, recursion and extensibility.

Objectives

Students are introduced to basic concepts of programming languages, including:

- Identify characteristics of procedural, object-oriented, functional, declarative, and scripting languages.
- Describe the phases of program translation.
- Explain different forms of binding, visibility, scoping, and lifetime management.
- Demonstrate the differences between call-by-value and call-by-reference parameter passing.
- Explain the concepts of encapsulation, abstraction, inheritance, and polymorphism.
- Write small programs in languages based on several different programming paradigms.

- Evaluate a language on the basis of the various features it supports.

Students also experience working on a team, developing a website, and giving an oral presentation.

Prerequisites

CS 321	Data Structures (corequisite)

In addition, familiarity with Unix, C, and Java is assumed.

Textbook and Other Resources

The textbook is:

> *Programming Language Pragmatics*, by Michael L. Scott. Third Edition. Morgan Kaufmann, 2009 ISBN: 9780123745149.

Activities

Grades are based on student performance of several activities. They are listed below, along with their relative weights.

Textbook Assignments	12%
Language Assignments	25%
Interpreter Assignments	15%
Language Website	18%
Exam	15%
Final	15%

Textbook Assignments (TA)

Several problem sets are assigned, from the exercises at the end of each chapter of the textbook. Students work on these individually, not as teams.

Language Assignments (LA)

Several small programs are assigned, to be developed in what are expected to be unfamiliar programming languages (e.g., C#). Open-source translators for these languages are available on the Linux computers in the Computer Science lab. Students work on these individually, not as teams.

Interpreter Assignments (IA)

A couple of programs are assigned, to extend a provided Java implementation of a simple programming-language interpreter. A Java development environment is available on the Linux computers in the Computer Science lab. Students work on these individually, not as teams.

Language Website (LW)

Each team of students develops a website dedicated to a particular, unfamiliar, programming language. Teams are formed, and languages are assigned, randomly. Several milestones are assigned. Open-source translators for these languages are available on the Linux computers in the Computer Science lab. Results are shared in an team-delivered oral presentation. Of course, students work in teams.

Exam and Final

An exam and a final are administered. These are in-class, open-note, and open-textbook (but no other books) tests. Of course, students work on these individually.

Documentation Standards

Good documentation and programming style is very important. Your programs must demonstrate these qualities for full credit. Good documentation and programming style includes:

- heading comments giving: author, date, class, and description
- function/procedure comments giving description of: purpose, parameters, and return value
- other comments where clarification of source code is needed
- proper and consistent indentation
- proper structure and modularity

When you submit a program, include: the source code, sample input data, and its corresponding results.

Grading

Homework is to be submitted at the beginning of lecture on the day it is due. Late work is typically not accepted. Try to let me know, ahead of time, if you will miss an examination.

Scores are posted near my office, as they become available. You are encouraged to check your scores to ensure they are recorded properly.

Academic Honesty

The following quotation is from the BSU Undergraduate Catalog. You should read that section.

> *The university's goal is to foster an intellectual atmosphere that produces educated, literate people. Because cheating and plagiarism are at odds with that goal, they shall not be tolerated in any form. Therefore, all work submitted by a student must represent that student's own ideas and effort; when the work does not, the student has engaged in academic dishonesty.*

There is related material in the BSU Student Handbook.

> *The course instructor is responsible for handling each case of academic dishonesty in the classroom except where a major or repeated offense is involved. In a proven case of cheating a student will be dismissed from the class and a failing grade issued.*

There are many forms of academic dishonesty. Some relevant examples include:

- Submitting programs, or parts of programs, written by someone else.
- Viewing exam answers, homework answers, or programs written by someone else. This includes material from other courses and previous semesters.
- Distributing exam answers, homework answers, or programs to someone else, even after it has been graded.

The BSU Undergraduate Catalog contains more examples. If you are unsure about a particular case, ask your instructor,

On homework, a student must work independently. Ideas and general principles can be discussed with other students, but work must be original.

Keep your files to yourself. See the Unix commands `chmod go-rwx` and `ls -l`.

On exams, of course, each student must work entirely independently.

Computer Accounts

Each student receives an account on the cluster of computers in the Computer Science Lab (ET-213/214), which runs the Linux operating system.

Schedule

Week	*Date*	*Topic*	*Assigned*	*Due*	*Reading*
1	Aug 26 Mon	Introduction			1
	Aug 28 Wed				
2	Sep 02 Mon	Labor Day			
	Sep 04 Wed				
3	Sep 09 Mon	Programming Language Syntax			2.0-2.1
	Sep 11 Wed		LA1,TA1		
4	Sep 16 Mon	Names, Scopes, and Bindings			3
	Sep 18 Wed		IA1		
5	Sep 23 Mon				
	Sep 25 Wed		LA2,TA2	LA1,TA1	
6	Sep 30 Mon				
	Oct 02 Wed		IA2	IA1	4.0-4.1
7	Oct 07 Mon	Control Flow			6
	Oct 09 Wed		LA3	LA2,TA2	
8	Oct 14 Mon				
	Oct 16 Wed				
9	Oct 21 Mon	Data Types			7
	Oct 23 Wed	Exam	LW1,LA4	LA3	
10	Oct 28 Mon				
	Oct 30 Wed			IA2	
11	Nov 04 Mon				
	Nov 06 Wed		LA5	LA4	
12	Nov 11 Mon	Subroutines and Control Abstractions			8.0-8.3
	Nov 13 Wed		TA3	LW1	
13	Nov 18 Mon				
	Nov 20 Wed			LA5	
14	Nov 25 Mon	Thanksgiving			
	Nov 27 Wed	Thanksgiving			
15	Dec 02 Mon	Presentations			
	Dec 04 Wed	Presentations			
16	Dec 09 Mon	Presentations			
	Dec 11 Wed	Presentations			
17	Dec 16 Mon	Final: 12:00-2:00			