CS 354: Programming Languages

Instructor

Instructor:	Jim Buffenbarger
Office:	CCP-359
Email:	`buff@cs.boisestate.edu`
Phone:	208-426-3567
WWW:	`http://cs.boisestate.edu/~buff`

BSU COVID-19 Response

Meetings

Lectures:	TuTh 1:30–2:45	CCP-221
Office hours:	MoWe 4:15–5:15	Zoom
	by appointment	Zoom

Zoom Lectures:	963 4792 6575
Zoom Office Hours:	938 4451 6635

Our Teaching Assistants / Graders are Michael Green and Emma Lytle:

```
michaelgreen1@u.boisestate.edu
emmalytle@u.boisestate.edu
```

CS Tutoring Center office hours can be found at:

```
onyx:~jbuffenb/classes/354/pub/TutorOfficeHours
```

I am happy to answer questions by email:

http://cs.boisestate.edu/~buff/files/EmailQuestions.pdf

Catalog Description

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

PREREQ: CS 321.

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

Goals

At the end of the course, the student will be able to do the following:

- identify characteristics of procedural, object-oriented, functional, and scripting languages
- describe the phases of program translation
- explain different forms of binding, visibility, scoping, and lifetime management
- demonstrate the differences between various parameter passing methods
- explain the concepts of encapsulation, abstraction, inheritance, and polymorphism
- write programs in languages based on several different programming paradigms
- evaluate a language on the basis of the various features which it supports

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

Textbook

- *Programming Language Pragmatics*, Michael L. Scott, Fourth edition, Elsevier: Morgan Kaufmann, 2015, ISBN: 9780124104099.

Other Course Material

This syllabus, lecture slides, assignments, and other material is available on the computers in the Computer Science Labs (CCP-240, CCP-241, and CCP-242), served by `onyx.boisestate.edu`, which is remotely accessible, via Secure Shell (SSH). It is *not* on the WWW, Blackboard, or elsewhere. It is in what is called our "`pub`" directory:

```
onyx:~jbuffenb/classes/354/pub
```

You may find the following local guide useful:

```
onyx:~jbuffenb/classes/354/pub/etc/cs-linux.pdf
```

Grading

At the end of the course, a letter grade is assigned to each student according to rank among classmates, which is determined from numerical scores assigned for performance of these activities:

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

Homework is due at 11:59PM, Mountain Time, on the day it is due. Late work is not accepted. To submit your solution to an assignment, login to a lab computer, change to the directory containing the files you want to submit, and execute:

`submit jbuffenb` *class assignment*

For example:

```
submit jbuffenb cs101 hw1
```

The `submit` program has a nice `man` page.

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

Scores are posted near my office, as they become available. You are encouraged to check your scores to ensure they are recorded properly. If you feel that a grading mistake has been made, contact me within two weeks of the date that work is returned.

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 programs are assigned, to be developed in what are expected to be unfamiliar programming languages (e.g., Scheme). 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.

Makeup examinations are not normally administered.

Source-Code Documentation

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

For more information, and examples, see:

www.cs.swarthmore.edu/~newhall/unixhelp/c_codestyle.html

Academic Integrity

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, those actions shall not be tolerated in any form. Academic dishonesty includes assisting a student to cheat, plagiarize, or commit any act of academic dishonesty. Plagiarism occurs when a person tries to represent another person's work as his or her own or borrows directly from another person's work without proper documentation.

If a student engages in academic dishonesty, the student may be dismissed from the class and may receive a failing grade. Other penalties may include suspension or expulsion from the University.

Much more information about academic integrity, including examples of academic dishonesty, is at:

http://cs.boisestate.edu/~buff/files/www-integrity.pdf

If you are unsure about a particular behavior, ask your instructor.

Labs and Safety

Each student receives an account on the cluster of computers in the Computer Science Labs: CCP-240, CCP-241, and CCP-242. The cluster comprises a server named `onyx.boisestate.edu` and a set of nodes with shared home directories. It is remotely accessible, via SSH. The cluster runs the Linux and Windows operating systems, via VMware.

Physical access requires building and room access. After-hours building access, and all-hours room access, require an authenticated proximity-type student-identification card.

You are responsible for understanding and obeying lab rules:

https://www.boisestate.edu/coen-its/labs/lab-rules/

Schedule

Week	*Date*	*Topic*	*Assigned*	*Due*	*Reading*
1	Jan 12 Tue	Introduction			1
	Jan 14 Thu				
2	Jan 19 Tue	Programming Language Syntax			2.0-2.1
	Jan 21 Thu				
3	Jan 26 Tue		TA1		
	Jan 28 Thu				
4	Feb 02 Tue		LA1		
	Feb 04 Thu				
5	Feb 09 Tue		IA1		3
	Feb 11 Thu				
6	Feb 16 Tue			LA1	
	Feb 18 Thu	Names, Scopes, and Bindings			
7	Feb 23 Tue	Control Flow			4.0-4.1
	Feb 25 Thu		LA2	TA1,IA1	
8	Mar 02 Tue		IA2		6
	Mar 04 Thu	Data Types	TA2		
9	Mar 09 Tue	Exam			
	Mar 11 Thu		LW1	LA2	7-8
10	Mar 16 Tue				
	Mar 18 Thu		LA3	IA2	
11	Mar 23 Tue				
	Mar 25 Thu	Subroutines and Control Abstractions		TA2	9.0-9.4
12	Mar 30 Tue				
	Apr 01 Thu		LA4	LA3	
13	Apr 06 Tue		TA3,LW2		
	Apr 08 Thu		LA5	LW1	
14	Apr 13 Tue	Spring Break			
	Apr 15 Thu	Spring Break			
15	Apr 20 Tue	Presentations		LA4,LW2	
	Apr 22 Thu	Presentations			
16	Apr 27 Tue	Presentations			
	Apr 29 Thu	Presentations		LA5,TA3	
17	May 04 Tue	Final: 2:30-4:30			