# CS 354: Programming Languages

## Instructor


| | |
|---|---|
| Instructor: | Jim Buffenbarger |
| Office: | CCP-258 |
| Voicemail: | (208) 426-3567 |
| Email: | mailto://buff@cs.boisestate.edu |
| WWW: | http://csweb.boisestate.edu/~buff |


## Meetings

| | | |
|---|---|---|
| Lectures: | TuTh 1:30–2:45 | CCP-243 |
| Office hours: | MoWe 11:45–1:15 | CCP-258 |
| | or by appointment | CCP-258 |

Lectures will be audio/video recorded, and made available, afterwards.

Our Teaching Assistant is Chris Dagher:

`ChristopherDaghe@u.boisestate.edu`

TA office hours are at (this "`pub`" directory is explained, below):

`pub/TutorOfficeHours`

I am happy to answer questions by email. Please please click here, or see:

`pub/doc/EmailQuestions.pdf`

## Catalog Description

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

PREREQ: CS 321.

## Goals

At the end of the course, students 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.

This year, new content has been added, as part of the *Semiconductors for All* initiative. New lectures, and a new homework assignment, have been incorporated, for the Verilog programming language. Verilog is a hardware-description language, for synthesizing semiconductor logic. More information is at the end of this syllabus.

## Textbook

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

## Other Course Material

This syllabus, lecture slides, assignments, and other material is available in what we'll called our "`pub`" directory. It is available in three places: GitHub, Canvas, and the `onyx` cluster of computers. This directory is read-only. So, you might want to copy it, perhaps to your local computer.

The GitHub `pub` directory can be accessed at GitHub, as shown in:

`pub/GitHub`

The Canvas `pub` directory can be accessed from our Canvas website, via the "Files" tab on the left sidebar.

The `onyx` `pub` directory can be accessed directly, by computers in our Computer Science Lab (CCP-240, CCP-241, and CCP-242).

Since `onyx.boisestate.edu` services Secure Shell (SSH) requests, you can also use SSH clients (e.g., `scp` and `sftp`) to access this `pub` directory remotely. However, beware: It contains symbolic links to parent directories, and `scp -r` will unconditionally follow them, thereby looping forever. To avoid this, use `sftp` or `tar`/`scp`, as needed.

In any case, the `onyx` `pub` directory is at:

```
~jbuffenb/classes/354/pub
```

The `onyx` cluster also has the advantage of containing all of the translators we will use.

## Grades

At the end of the semester, the Registrar requires a letter grade for each student. Accordingly, during the semester, homework and exams are evaluated, and numeric scores are assigned. Each such artifact is worth a certain number of points, and has a weight. From these scores, a student's overall numeric/raw percentage is computed.

Collectively, a class's raw percentages form a distribution, with a mean and standard deviation, sometimes called a Bell Curve. An algorithm normalizes these values into a "grading" distribution, with a desired mean and standard deviation of 85 and 10 (i.e., the values are "curved"). A normalized value is then mapped to a letter grade, in the conventional way. If you are interested in the gory details, click here.

During the semester all of this evolving data can be found in our `pub` directory, on `onyx`, its nodes, Canvas, and GitHub.

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 cs354 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 in our `pub/scores` directory, as they become available. You will receive a code, by email, indicating your row in the score sheet. 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 as soon as possible.

The weights of homework and exams is shown in the table below:

| *Activity* | *Weight* |
|---|---|
| Book Assignments | 10% |
| Language Assignments | 30% |
| Translator Assignments | 20% |
| Language Website | 20% |
| Exam | 10% |
| Final | 10% |

## Book Assignments (BA)

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

The rubric that will be used to grade each BA is distributed with the assignment.

## Language Assignments (LA)

Students are assigned several programs, to be developed in what are expected to be un-familiar 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.

The rubric that will be used to grade each LA is distributed with the assignment. Try to focus on the assignments, rather than the rubrics.

### Translator Assignments (TA)

Students extend a Java implementation of a simple programming-language translator (i.e., a combination interpreter/compiler). A Java development environment is available on the Linux computers in the Computer Science lab.

Students work on these individually, not as teams.

The rubric that will be used to grade each TA is distributed with the assignment. Try to focus on the assignments, rather than the rubrics.

### Language Website (LW)

Students are randomly partitioned into teams, of 3–5 members. Each team chooses an unfamiliar programming language, and develops a website dedicated to that language. Website content includes example programs of varying complexity. During the last couple of weeks of the semester's lectures, each team introduces their language, and describes the experience of programming in that language, in an in-person team-delivered oral presentation.

Of course, team members work together on this assignment.

The rubric that will be used to grade this assignment is distributed with the assignment, but scores will not be made available before the end of the semester.

### Exams

An exam and a final are administered. These are in-class, open-note, open-textbook, and open-laptop tests. Students work on these individually.

## Attendance

In-person lecture attendance is an important part of course participation. Attendance is taken at each lecture: starting five minutes before the scheduled start time, and ending fifteen minutes after the scheduled start time. Attendance is not taken during the first week of classes, holidays, or finals week.

Attendance can affect your grade. Each absence results in a one-percent reduction of your overall normalized percentage. Since a few absences are expected, completion of BSU's on-line end-of-semester course evaluation will erase up to five absences. Since evaluations are anonymous (at least to me), I will try to remind you to email me evidence of completing one: a screenshot will do.

Attendance is administered wirelessly, via the iClicker app, available for free, from your smartphone's app store. For more information, click here.

## 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, click here.

## 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.

For much more information about academic integrity, including examples of academic dishonesty, please click here. 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 operating system.

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.

## Semiconductors for All

Our second homework assignment is about a programming language named Verilog. Verilog is a hardware-description language, for synthesizing semiconductor logic. The development of the assignment, its solution, and the accompanying lecture material, was funded as part of a $5M grant to BSU, named *Semiconductors for All*. The grantor was the *Idaho Workforce Development Council*, the *Idaho Department of Labor*, and, ultimately, President Biden's $280B *CHIPS and Science Act*, of 2022.

The grant requires our syllabus include the following content. Please read it. I'll explain the "quiz" part, later.

> *The course you are enrolled in has been supported through the Semiconductor for All grant with the Institute for Microelectronics Education and Research at Boise State University. Our goal is to provide you with a high-demand set of skills and education that enhance your current job and prepare you for future jobs. The Semiconductor for All program was funded through a generous grant from Idaho Workforce Development Council, which helped make curriculum changes to this course.*
>
> *As part of the grant requirements, we need to receive your permission to share your participation in this course/program. The Idaho Workforce Development Council will use this information to track the success of their investment in the Semiconductor for All grant.*
>
> *With your permission, we will use an established and secure process to transfer your personal information to the Idaho Department of Labor. This includes your name, SSN, and student ID. Providing this information links your student information at Boise State University with your information at the Idaho Department of Labor*

*allowing verification of your employment and wage data to document the economic impact of each student's experience.*

*This process has been approved by the Chief Information Security Office of Boise State University. We will generate a list of student ID numbers with names that will be sent to the Boise State University Office for Institutional Effectiveness. Nobody but the Idaho Department of Labor will see your personal information. Your personal information will not be sold or used for anything but the purposes outlined here. Regardless if you choose to participate or not, once you complete the quiz, the first part of this course's content will be available to you.*

## Schedule

| *Week* | *Date* | *Topic* | *Assigned* | *Due* | *Reading* |
|---|---|---|---|---|---|
| 1 | Jan 14 Tue | Introduction | | | 1 |
| | Jan 16 Thu | Scheme | | | |
| 2 | Jan 21 Tue | Scheme | | | |
| | Jan 23 Thu | Introduction | LA1 | | |
| 3 | Jan 28 Tue | | | | |
| | Jan 30 Thu | | | | |
| 4 | Feb 04 Tue | | | LA1 | |
| | Feb 06 Thu | Verilog | BA1 | | |
| 5 | Feb 11 Tue | Verilog | | | |
| | Feb 13 Thu | Verilog | LA2 | | |
| 6 | Feb 18 Tue | Programming Language Syntax | | | 2.0-2.1 |
| | Feb 20 Thu | | | | |
| 7 | Feb 25 Tue | Names, Scopes, and Bindings | TA1 | LA2 | 3 |
| | Feb 27 Thu | | | | |
| 8 | Mar 04 Tue | Semantic Analysis | LA3 | BA1 | 4.0-4.1 |
| | Mar 06 Thu | Exam | | TA1 | |
| 9 | Mar 11 Tue | | TA2 | | |
| | Mar 13 Thu | Control Flow | | | 6 |
| 10 | Mar 18 Tue | Spring Break | | | |
| | Mar 20 Thu | Spring Break | | | |
| 11 | Mar 25 Tue | | BA2 | | |
| | Mar 27 Thu | Data Types | | | 7-8 |
| 12 | Apr 01 Tue | Subroutines and Control Abstractions | LW | | 9.0-9.4 |
| | Apr 03 Thu | | | LA3 | |
| 13 | Apr 08 Tue | | LA4 | TA2 | |
| | Apr 10 Thu | | BA3 | BA2 | |
| 14 | Apr 15 Tue | | | | |
| | Apr 17 Thu | | LA5 | LA4 | |
| 15 | Apr 22 Tue | LW Presentations | | LW,BA3 | |
| | Apr 24 Thu | LW Presentations | | | |
| 16 | Apr 29 Tue | LW Presentations | | LA5 | |
| | May 01 Thu | LW Presentations | | | |
| 17 | May 06 Tue | Final 2:30-4:30 | | | |