

---

$\preceq$  COMP 335 Principles of Programming Languages  $\succeq$   
*M-W-F Lecture 10:30-11:20 (DC 1315 – csLab)*

---

Who: Michael Gousie  
Where: Discovery Center 1325  
When: Mon, Wed, Fri 11:30-12:30; Tue 1:30-3:00  
and by appointment  
E-mail: mgousie(at)wheatoncollege(dot)edu  
Web: <https://cs.wheatoncollege.edu/mgousie>

---

**Content:**

This is not *primarily* a course to teach you how to program in various languages, although you will write some interesting software. Instead, this course provides an introductory theoretical study of the principles that govern the design and implementation of programming languages. To this end, we will examine language syntax using tools such as BNF notation and parsing. We will delve into programming language constructs and compare how different languages implement those constructs. Finally, we will look at different language paradigms, comparing procedural, functional, object oriented, and logic programming languages.

The course is split between theoretical ideas and practical applications. There will be written homework assignments taken from programming language texts and other sources, as well as projects in different programming languages, such as Bash, C, Scheme/Lisp, and Prolog. A major component of the course is the coverage of Java.

**Objectives:**

At the completion of the course, students will be able to:

- Define a programming language's syntax using BNF notation
- Write system code in both Bash and C
- Write basic code in functional and logic languages
- Write more effective code by utilizing programming language features/quirks

**Required Text:**

- Sebesta, Robert. *Concepts of Programming Languages*, 12<sup>th</sup> Edition. (Pearson 2019).
- Additional resources will be available on the course web page.

**Requirements:**

There will be two exams during the semester and a final exam. The exams will comprise 50% of your grade. You will write five programs in different languages for a total of 45%. Because the programming projects will be of varying length and difficulty, they will not be evenly weighted (see below). Some projects can be done in small groups; the details will be explained in each program specification. Several of the programs will be written on the Linux machines using compilers/interpreters available or freely obtainable or using a suitable laptop (running Linux or Mac OS). The remaining 5% will be the average of a few written homeworks and/or short programs.

**Grading:**

Grades will be assigned according to the following scale:

A = 93-100, A- = 90-92, B+ = 87-89, B = 83-86, B- = 80-82, C+ = 77-79, etc.

**Exam Schedule:**

Exam	Weight	Date
Exam 1	14%	October 6
Exam 2	18%	November 7
Final Exam	18%	December 12 @ 9:00 AM

**Assignment Schedule:**

Program	Weight	Language Type	Topic (Subject to change)	Duration (Approx.)
PL1	5%	Scripting	Bash scripts	1 week
PL2	8%	System Tools	A little C	2 weeks
PL3	20%	Procedural	Parsing using C	4 weeks
PL4	7%	Functional	Scheme	2 weeks
PL5	5%	Logic	Prolog	1 week

**Course Policies:**

- You are responsible for all material covered in class.
- You are responsible for all reading as assigned/handed out in class. There will be significant material needed for projects that is not covered in class.
- Programming projects will be coded in various languages. All compilers/interpreters will be available on Linux (because they're all FREE!) in csLab, although you may use any platform and software you wish. However, the projects will be tested on Ubuntu Linux using the tools described in class, so at the very least you should test your finished projects in csLab/Linux.
- If you must miss a quiz or exam for any reason, you must inform me **before** the test. Except in the case of emergency, illness (almost death), or you've fallen off one of Wheaton's spires, makeup exams will not be given.
- Written homeworks should be neat and done on loose-leaf or plain paper. Program code as part of a written homework should be typed. Do not tear pages out of a notebook so the resulting paper has "fringes" on it. Staple multiple pages together.
- Assignment due dates are **firm**.
  - All programming projects must be submitted in Canvas by 11:59:59 PM on the due date unless otherwise noted. Projects submitted on the following day will receive a 15% penalty. Anything turned in later will receive a 0. Hard copy, if required, must be submitted at the beginning of class on the next day or as instructed on the specification sheet.
  - Written homeworks/papers must be submitted at the beginning of class on the due date. There is no provision for homework turned in late.

- You are expected to adhere to the Honor Code.
  - Although *discussion* of projects or homework is encouraged, the *implementation* of programs is to be the result of your **own** or your group's work.
  - AI can help you speed up the programming process by having it do some of the more menial tasks. However, your program should still be **your own work**. Copy/paste is not the way to learn how to program, whether you are doing this from another person or from an AI application. If you are unsure where the line is between collaborating with AI and copying from AI, we recommend the following heuristics:
    - \* Never hit *Copy* within your conversation with an AI assistant (and then *Paste* into your code). You can copy your own work into your conversation, but do not copy anything from the conversation back into your assignment. Instead, use your interaction with the AI assistant as a learning experience, then let your assignment reflect your improved understanding.
    - \* Do not have your assignment and the AI agent itself open on your device at the same time. Similar to above, use your conversation with the AI as a learning experience, then close the interaction down, open your assignment, and let your assignment reflect your revised knowledge. This heuristic includes avoiding using AI assistants that are directly integrated into your composition environment: just as you should not let a classmate write content or code directly into your submission, so also you should avoid using tools that directly add content to your submission.
  - If a project looks suspicious, I may ask you to explain the purpose, function, and details of your code; if you can't, it will be considered plagiarized.
  - Collaboration on exams is prohibited.
  - Any violation of the above guidelines will result in a 0 for the assignment/exam and/or a failing grade for the course.
  - You will be required to write and **sign** the pledge on all work turned in: *I have abided by the Wheaton Honor Code in this work*. On web/site pages, this should be included in a comment at the top of each file or as directed on the assignment specifications.
- Unless specified, the use of a laptop or other computer/tablet is **not allowed** during lecture. Special arrangements may be made for use of a laptop/tablet during lectures, if necessary.
- The use of cell phones, iPods, iPads, iPlops, iFlops, and other personal electronic devices is prohibited during class, labs, and exams.
- Once class begins, no one should leave class except in the case of an emergency.
- Accommodations for disabilities:

*Wheaton College is committed to providing equitable access and supportive services for all students to fully access and thrive in the academic, residential and social aspects of student life. Affirmatively, Wheaton provides appropriate accommodations for eligible students with documented disabilities to afford equal access to educational programs and services. Individuals with disabilities and other access concerns requiring accommodations or information on accessibility should reach out to Accessibility Services at the Filene Center:  
~ accessibility@wheatoncollege.edu or (508) 286-3794 ~*

**Course Schedule (subject to change):**

Wk #	Week Begin	Topic	Reading
1	Aug 24	<b>Introduction</b>  What are we doing? Linux	Chs. 1 & 2,  Handouts
2	Aug 31	<b>System Programming</b>  <i>No class Sep 1 (Labor Day)</i> Bash Scripting	Online resources; handouts
3	Sep 7	<b>System Programming</b>  More Bash Programming in C	Online resources and/or text on C
4	Sep 14	<b>C</b>  Parameter passing/pointers Compiler basics	Online resources and/or text on C
5	Sep 21	<b>Syntax</b>  BNF notation Lexical analysis (Scanning)	Ch. 3
6	Sep 28	<b>Syntax Analysis</b>  Parsing concepts Parsing methods	Ch. 4
7	Oct 5	<b>Compare-n-Contrast</b>  Exam 1 (Monday) Binding Scope	Ch. 5; handouts
8	Oct 12	<b>Types and Expressions</b>  <i>No class Oct. 13 (October Break)</i> Type conversions Relational expressions Short-circuit evaluation Assignment statements <i>MAP Day October 16</i>	Selections, Chs. 6 & 7

Wk #	Week Begin	Topic	Reading
9	Oct 19	<b>Control</b>  Selection Iteration	Ch. 8
10	Oct 26	<b>Functional Programming</b>  Lisp Scheme	Ch. 15
11	Nov 2	<b>Subprograms</b>  Basics Parameter passing Overloading Exam 2 (Friday)	Ch. 9
12	Nov 9	<b>More Subprograms</b>  Recursion System stack Nested subprograms	Ch. 10
13	Nov 16	<b>Logic Programming</b>  Prolog Brain implosion	Ch. 16
14	Nov 23	<b>Exception Handling</b>  Exceptions in C++ <i>No class Nov. 26/28 – Thanksgiving</i>	Ch. 14; Cranberry Sauce
15	Nov 30	<b>OOP</b>  C++ vs. Java/Python	Ch. 12
16	Dec 7	<b>Final Exam Week</b>  Final Exam on December 12 @ 9:00 AM – oof!	