# Principles of Programming Languages(PoPL)

Summer 2019

| | | | |
|---|---|---|---|
| **Instructor:** | John Martin (Jack) | **Time:** | T,Th 2:00PM – 3:15PM |
| **Email:** | John.P.Martin@colorado.edu | **Place:** | FLMG 155 |

## Contents



## 1 Course Pages

1. Notion
   This is the wiki for the course and all information for the class will be placed here. This includes a calendar, homework, and references/resources.
2. `https://moodle.cs.colorado.edu`
   We will use moodle for grading and homework submissions. We may change this to Canvas so stay posted.

## 2 Staff & Office Hours

**Guidelines on Contacting Staff** When contacting the course staff please first consider the proper member of the staff to contact given your problem.

- For course help, tutoring, material clarification, please first contact the Course Assistants and then reach out to the Teaching Assistants.
- For logistics around Exams please inform your Teaching Assistant and they will inform the Instructor.
- For Accommodations, Absences, or other Instructor-Level questions, concerns, etc. please reach out to Jack directly.

Please try to follow these guidelines closely. There are 200 students in the course this fall so contacting the incorrect people for help could result in delayed responses. Please respect our time and we should be able to keep everything running smoothly.

**For Staff Contact and Office Hour Information please see the course wiki(refer to the links above.)**

## 3 Main References

- The primary readings for the course will be the wiki pages for the topics being presented in lecture in a given week. These will be added as the course goes on as we are building this out for the first time. Most of the pages we write will include external resources. These will be Wikipedia entries, tutorials, and high-quality blog posts which we feel convey the information well. There is fortunately a large community in this area of study and they seem to enjoy writing.

- The course follows many ideas from a supplemental text: Essentials of Programming Languages, 3rd edition by Daniel P. Friedman and Mitchell Wand. This book is available as an e-book to all University students, but not required.
- We also strongly recommend that you get access to Programming in Scala, **third edition** by Martin Odersky, Lex Spoon, and Bill Venners. This book is an extended tutorial for learning Scala by those directly involved in the language's development. This book is available online through our library. The use of the **third edition** is important.

# 4 Introduction

Programming languages are by no means perfect. You have undoubtedly found yourself asking why your program is behaving in a strange way or spent hours trying to squash a bug that in hindsight was frustratingly trivial. This only adds to the anger when you realize you forgot what the other half of the program did in the first place. **What if there was a way to avoid these kinds of bugs in the first place?**

After that it seems like a good time to go tweak another section of the program. Oh, whoops, you also forgot to document that half. There is always time in the future to begin writing the novel required to make sense of what your past self thought was a clever solution to a problem. Thus we come to a Commandment of Programming:

> Programs must be written for people to read, and only incidentally for machines to execute.
>
> Harold Abelson
> *Structure and Interpretation of Computer Programs*

It is all too easy to forget this little fact. Deep down, every programmer knows it and may even make a cursory attempt at following it, but this is rarely enough. **What if there was a better way to write programs so that I could make sense of them later?**

You consult a rebellious friend who raves about arcane subjects such as **tail recursion**, **algebraic types**, **monads**. This strange fellow recommends a language that could help you with some of the problems you've been having. **But what is the best way to learn a new language?** And more importantly **What are the principles that underlie all programming languages?**

> There are many ways of trying to understand programs. People often rely too much on one way, which is called "debugging" and consists of running a partly-understood program to see if it does what you expected. Another way, which ML advocates, is to install some means of understanding in the very programs themselves
>
> Robin Milner
> *Creator of ML &*
> *Discoverer of the Hindley-Milner Type System*

In this course I will try to answer the questions that the (imaginary) student in my head posed above. We will talk about the deep, foundational, and beautiful ideas that underlie type theory, the *means of understanding* that Robin Milner alluded to above. We'll explore concepts that will make you think about programming languages in a new light and, I hope, change your relationship towards programming.

Over the course of the semester we will learn about programming languages in two ways. The first is by learning a multi-paradigm language called Scala. This language takes the best of object-oriented and functional programming to create a production-quality system for writing programs. We will use Scala to implement several of our own languages.

We will also study some simple formal systems, espsecially towards the beginning of the course. These are the Peter Pans of the PL world: they are forever young, doing a job well and changing very little. We'll learn interesting secrets from these deceptively simple collections of rules. Surprising secrets that we may realize we have known all along.

That is all I'll say for now. I'll look forward to the journey.

# 5 Course Work

**Weekly Problem Sets** Each week, you will receive a problem set that will include concept-based problems and programming assignments in Scala. These assignments will be posted on Moodle as Jupyter notebooks that need to be filled out by the students. We will be using the `nbgrader` auto-grader, so the accompanying submission instructions must be followed precisely.

**Midterms** We will have at least five midterm exams that will be conducted during your recitations. These exams will last roughly 30 minutes and consist of material covered in class or through your assignments.

**Final Exam** A final exam will account for 20% of the grade. This will be at a time announced by the University registrar. The final is compulsory for all students desiring a passing grade in this class.

| | |
|---|---|
| Weekly Problem Sets | 40% |
| Midterms | 40% |
| Final Exam | 20% |

Figure 1: Grading Breakdown

# 6 Collaboration Policy

You are welcome and encouraged to work together in learning the material. However, whatever you submit must be your own. In other words, cutting and pasting or copying verbatim from another source be it a classmate, an online source or even something that the TA/instructor showed you is strictly forbidden.

**Cite Your Sources** If you worked with someone on an assignment or your submission includes quotes from a book, paper, or web site, then you should clearly acknowledge the source. Bottom line, feel free to use resources that are available to you as long as the use is reasonable and you cite them in your submission. Copying answers directly or indirectly from solution manuals, web pages, or your peers is, however, forbidden.

**Inspiration is free** You may discuss homework assignments with anyone. You are especially encouraged to discuss solutions with your instructors and classmates.

**Plagiarism is forbidden** Any assignments and code that you turn in should be written entirely by you. You should not need to consult sources beyond your textbook, class notes, posted lecture slides and notebooks, programming language documentation, and online sources for basic techniques. Copying/soliciting a solution to a problem from the internet or another classmate constitutes a violation of the course's collaboration policy and the honor code and will result in an F in the course and a trip to the honor council.

**Do not search for a solution online** You may not actively search for a solution to the problem from the internet. This includes posting to sources like StackExchange, Reddit, Chegg, etc.

**StackExchange Clarification** Searching for basic techniques in Python/Pandas/Numpy is totally fine. If you want to post and ask "How do I group by two columns, then do something, then group by a third column" that's fine. What you cannot do is post "Here's the problem my prof gave me. I need to convert Age in Earth years to Martian years and then predict the person's favorite color. Give me code!". That's cheating.

**When in doubt, ask** We have tried to lay down some rules and the spirit of the collaboration policy above. However, we cannot be comprehensive. If you have doubts about this policy or would like to discuss specific cases, please ask the instructor. If it has not been described above, you should discuss it with us first

# 7 Important Dates

Start of Term ..... August 26th, 2019
Fall Break (No Class) ..... November 24th - 30th, 2019
Last Lecture ..... December 12th, 2019
Final Exam ..... 4:30pm December 16th, 2019

# 8 Class Policy

**Regular attendance** is essential and expected. I teach in a Socratic style, so missing class means missing certain explanations or models that are not always present in the text. Attendance will be taken daily.

**No laptops or electronic devices in class** This will be a course mostly taught ye ole-fashioned way, using the chalkboard. As such, please bring note-taking material to each class. Fortunately, the material is very conducive to being handwritten.

**Recitations** The exception to the above rule is for the recitations which will be on Friday of each week. These will be when we do the bulk of the instruction in Scala and the finer details of implementation.

# 9 Topics by Category

## 9.1 FUNCTIONAL PROGRAMMING in Scala

- Recursion and Recursive Functions
- Case Classes
- Pattern Matching
- `fold`, `map`, `filter`

## 9.2 CONTROL FLOW

- Continuations
- Continuation Passing Style
- Models for Concurrency and Concurrent Programming

## 9.3 SEMANTICS & INTEPRETERS

- Grammars and Inductive Definitions
- Big-Step (Operational) Semantics
- Lettuce: A Functional Programming Language with Let Bindings
- Scoping Rules
- Implementing interpreters with Functional Calls
- Recursion in Lettuce
- References and State in Lettuce
- Implicit References(`val`/`var`)
- Type Systems: Using Type Theory to Build Programs
- Type Inference

## 9.4 Parsing

- Types of Parsers
- Parser Combinators
- Parsing Lettuce

## 9.5 TYPE THEORY

- Introduction to Types
- Algebraic Datatypes
- Parametric Polymorphism, Abstract Datatypes, and Generics
- Introduction and Elimination

## 9.6 ADVANCED TOPICS

- Imperative Programming Semantics
- Dependent Types
- Probabilistic Programming Languages
- Differentiable Programming

# 10 University Course Policies

## 10.1 Honor Code

All students enrolled in a University of Colorado Boulder course are responsible for knowing and adhering to the academic integrity policy of the institution. Violations of the policy may include: plagiarism, cheating, fabrication, lying, bribery, threat, unauthorized access, clicker fraud, resubmission, and aiding academic dishonesty. All incidents of academic misconduct will be reported to the Honor Code Council (honor@colorado.edu; 303-735-2273). Students who are found responsible for violating the academic integrity policy will be subject to nonacademic sanctions from the Honor Code Council as well as academic sanctions from the faculty member. Additional information regarding the academic integrity policy can be found at www.colorado.edu/honorcode/.

## 10.2 Disability Accommodations

If you qualify for accommodations because of a disability, please submit your accommodation letter from Disability Services to your faculty member in a timely manner (for exam accommodations provide your letter at least one week prior to the exam) so that your needs can be addressed. Disability Services determines accommodations based on documented disabilities in the academic environment. Information on requesting accommodations is located on the Disability Services website. Contact Disability Services at 303-492-8671 or dsinfo@colorado.edu for further assistance. If you have a temporary medical condition or injury, see Temporary Medical Conditions under the Students tab on the Disability Services website and discuss your needs with your professor.

## 10.3 Religious Observances

Campus policy regarding religious observances requires that faculty make every effort to deal reasonably and fairly with all students who, because of religious obligations, have conflicts with scheduled exams, assignments, or required attendance. If you have an exam or assignment conflict due to a religious observance please notify your instructor in a timely manner. See the campus policy regarding religious observances for full details.

## 10.4 Classroom Behavior

Students and faculty each have responsibility for maintaining an appropriate learning environment. Those who fail to adhere to such behavioral standards may be subject to discipline. Professional courtesy and sensitivity are especially important with respect to individuals and topics dealing with race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Class rosters are provided to the instructor with the student's legal name. We will gladly honor your request to address you by an alternate name or gender pronoun. Please advise me of this preference early in the semester so that I may make appropriate changes to my records. For more information, see the policies on classroom behavior and the Student Code of Conduct.

## 10.5 Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

The University of Colorado Boulder (CU Boulder) is committed to maintaining a positive learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct, discrimination, harassment or related retaliation against or by any employee or student. CU's Sexual Misconduct Policy prohibits sexual assault, sexual exploitation, sexual harassment, intimate partner abuse (dating or domestic violence), stalking or related retaliation. CU Boulder's Discrimination and Harassment Policy prohibits discrimination, harassment or related retaliation based on race, color, national origin, sex, pregnancy, age, disability, creed, religion, sexual orientation, gender identity, gender expression, veteran status, political affiliation or political philosophy. Individuals who believe they have been subject to misconduct under either policy should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127. Information about the OIEC, the above referenced policies, and the campus resources available to assist individuals regarding sexual misconduct, discrimination, harassment or related retaliation can be found at the OIEC website.