

CSC 347 - Concepts of Programming Languages

Course Overview

Instructor: Stefan Mitsch



Programming Languages

- **1970:** assembly, FORTRAN, COBOL, Lisp
- **1980:** C, Pascal, BASIC, ML, Smalltalk
- **1990:** C++, Perl, Objective C, Erlang
- **2000:** Java, JavaScript, Python, Ruby, Lua
- **2005:** C#
- **2010:** Scala, F#, Clojure, Go
- **2015:** Rust, Swift, Kotlin, Elm, Elixir, TypeScript, PureScript
- **2020:** ReasonML, Crystal, Pony, Zig
- Many more!



Programming Languages

- Programming languages keep popping up
- You will have to keep learning new languages
- And keep up with changes to current languages
 - Java 8 and C++ 11 added Lambda expressions (nested, anonymous functions)
- 💡 Lots of common concepts!



Learning Objectives

Learn PLs more easily by recognizing concepts

- "it has while loops"
- "it has closures"
- "it has list comprehensions"
- "it has dynamic dispatch"
- Deeper understanding of PL concepts / paradigms
- Impact of PL on program development, modularity, correctness, runtime efficiency, etc.



Non-Goals

- Learn a PL similar to one you already know
- Instead, learn concepts to facilitate quick assimilation of new PLs as they appear
- Learn an IDE
- Instead, become familiar with a range of tools and build systems



Course Overview

❓ What are different ways of expressing computations?

- **Programming paradigms and styles:**
 - functional vs object-oriented
 - mutability vs immutability
 - iteration vs recursion
 - pattern matching vs visitor design pattern
- **Runtime Storage:** lexical and dynamic scope; stack layout; how to support inheritance and dynamic dispatch; nested structures (functions or objects); garbage collection
- **Static Analysis:** dynamic vs static type checking; subtyping; parametric polymorphism



Course Approach

- Hands on: write many programs and experiments
- **Scala** as main language:
 - carefully designed *multi-paradigm* language
 - textbook explains PL concepts in context
- Also bits of: C, C++, C#, Java, JavaScript...
 - ... Kotlin, Swift, Rust, Scheme, Haskell
 - chosen as exemplars of concepts



Discussion Forum

- On D2L
- Use appropriate language

Asking Questions

- For non-personal messages use D2L
 - "I think there is a mistake with grading of Question 3 of Homework 2." - **direct message/email to instructor**
 - "I cannot run program `foo`. I have tried running it from the command line on OS X. See below for a transcript of what I typed and the error message I received. Could you please help?" - **ask in D2L discussion forum**



Asking Questions

- Include name, student number, and class in messages directly to the instructor
- Include enough context to answer your question:
 - "program `foo` doesn't work." - *bad*
 - "program `foo` fails with output pasted below" - *good*
 - "program `foo` fails with input and output pasted below" - *better*
 - "program `foo` fails with input and output pasted below; using OS X; I have run extra commands to show current working directory, the version of `foo` in use, and other relevant information" - *best*



Course Syllabus

- Review the [Syllabus](#) linked from the [course homepage](#)
- Programming in Scala, First Edition is available for free [online](#)
- **Get Programming in Scala 5th edition**
- Earlier editions use a different version of Scala than class; you can use them but may need to look up new syntax