Spring 2024 Semester

CS 320: Principles of Programming Languages

Introduction, Class Logistics, Course Objectives

Teaching Staff

- *Instructors:* **Assaf Kfoury** and **Nathan Mull**
- *Teaching Fellows:* **Zachery Casey** and **Qiancheng (‘Robin’) Fu**
- *Teaching Assistants:* **Jason Wang** and **Sebastian Wu**

More information about the course structure and its logistics are at the (public) website of the course. Click here .

To retrieve homework assignments and lecture materials, you will need to go to the (public) GitHub repository of the course. Click here .

Homework assignment HW0 is posted today in the GitHub repository, due one week from today (January 25, by 11:59 pm)

Brief History of Programming Languages

- How many programming languages have been invented since the beginning of *Computer Science* as a field of study? Perhaps 100? Perhaps 200?

Brief History of Programming Languages

- How many programming languages have been invented since the beginning of *Computer Science* as a field of study? Perhaps 100? Perhaps 200?
- How many programming languages have you come across? Perhaps 10? Perhaps 15?

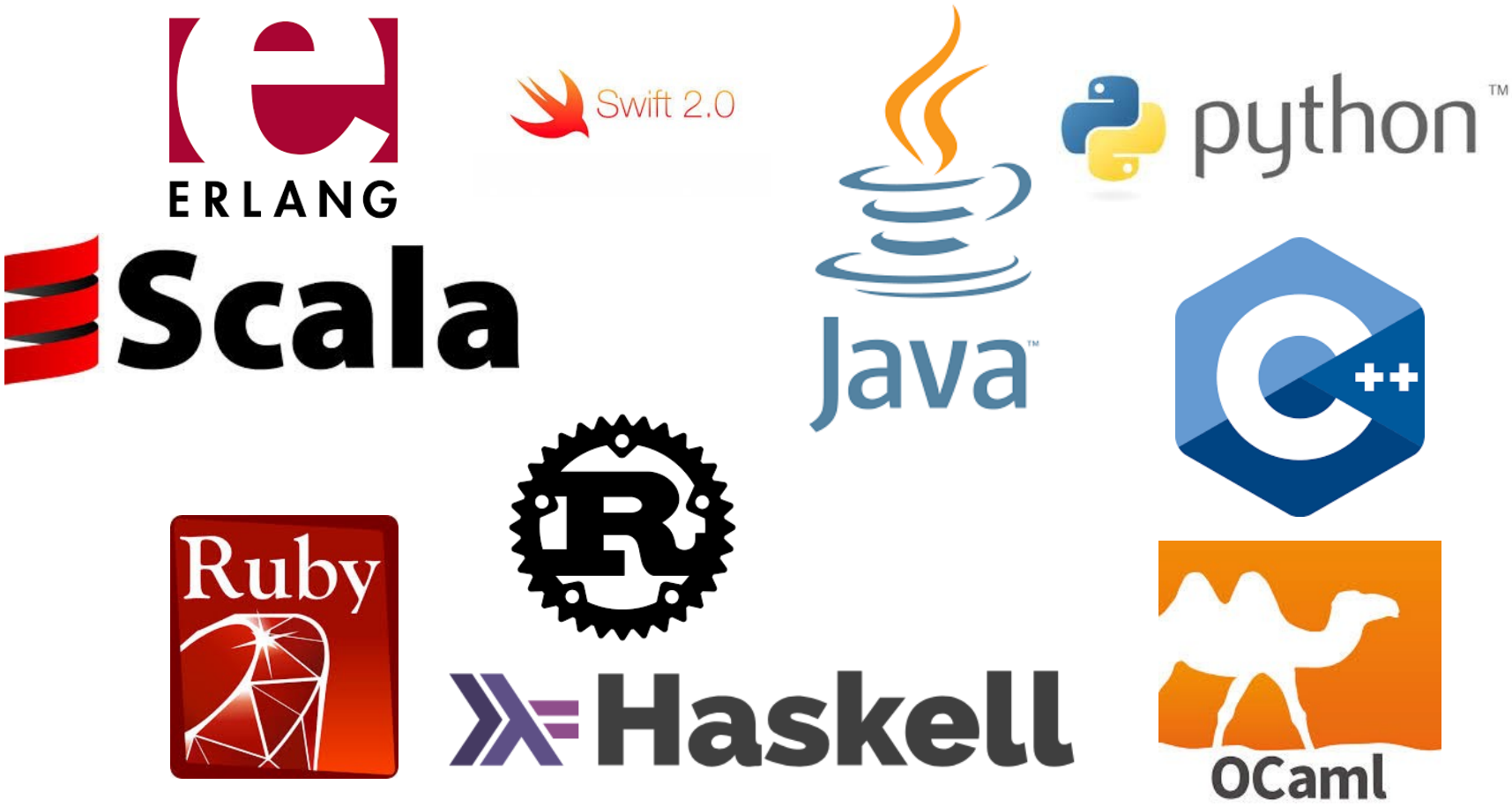
Brief History of Programming Languages

- How many programming languages have been invented since the beginning of *Computer Science* as a field of study? Perhaps 100? Perhaps 200?
- How many programming languages have you come across? Perhaps 10? Perhaps 15?
- How many programming languages have you learned and used? Perhaps 2? Perhaps 5?

Brief History of Programming Languages

- How many programming languages have been invented since the beginning of *Computer Science* as a field of study? Perhaps 100? Perhaps 200?
- How many programming languages have you come across? Perhaps 10? Perhaps 15?
- How many programming languages have you learned and used? Perhaps 2? Perhaps 5?
- For a brief history of programming languages, click here.

What is your favorite language?



Why is it your favorite?

- Easy to program?
- Good debugging environment?
- Large amount of standard libraries?
- Syntax?
- Language you know the most?
- Performance (or other runtime characteristics)?
- Validation mechanisms (type checking, verification)?
- Garbage collection?

What do we study in Programming Languages? PL

- Language Design
  - Programming Constructs, Abstractions
- Formal mechanisms to reason about and specify programs / languages
  - Type Systems, Verification
- Compiler Design
  - Optimizations, Program Analysis, JIT
- Language Runtime Design
  - Virtual Machines, Garbage Collection, JIT, Interpreters

What will we look at?

- We will study the functional programming paradigm
- We will study the different features and constructs of programming languages,
- We will look at more formal ways to compare different languages design choices,
- We will experiment with building our own language.

Topics Covered (tentative)

- Functional Programming
- Language Design and Evolution
- Syntax and Semantics
- Names, Bindings, and Scopes
- Data Types
- Expressions and Assignment Statements
- Control Structures
- Subprograms and their Implementation
- Exceptions and Event Handling (tentative)

Programming Language we will use

www.ocaml.org

Programming Assignments

- The solutions to the programming assignments must be your own. No pair or group submission is allowed.
- If you use for standard tasks some snippets of code taken from some source, e.g. github, stackoverflow, real world OCaml, etc. you need to add a comment saying so.
- Any violation will be considered a violation of academic integrity and reported to the school.

Assignment and Homework submissions

- Submission via *Gradescope*
- More details to come

Late Homework Submissions

- No late homework submissions
- Late homework will be given a 0 grade

Re-Grading of homework

- Re-grade requests must occur within one week of grades being released,always via **Gradescope** – not by email.
- Consult with your TF or TA first, if a disagreement still persists, come speak to one of the two instructors.

Structure of classes

- Functional Programming in OCaml (first few weeks)
- Concepts of Programming languages

What is a Functional Language

A functional language:

- defines programs in a way similar to the one we use to define mathematical functions,
- avoids the use of mutable states (states that can change) in describing what a program should do.

In a functional language, the information is maintained by the computation.

Thinking Functionally

In imperative languages like Java or C, you get most work done by changing the state of the memory, via variables or data structures

```
temp  = pair.x;
pair.x = pair.y;
pair.y = temp;
```

Commands modify or change the state – in this case an existing data structure (pair)

In functional languages you get most work done by producing something new

```
let
 (x,y) = pair
in
 (y,x)
```

Commands analyze an existing data (pair) and produce a new data (y,x)

Advantages of Functional Languages

Functional languages predict the future

- Garbage collection Java [1995], LISP [1958]
- Generics Java 5 [2004], ML [1990]
- Higher-order functions
  C#3.0 [2007], Java 8 [2014], LISP [1958]
- Type inference
  C++11 [2011], Java 7 [2011] and ML [1990]
- What's next?

Functional languages are more and more used in industry

- Java 8 ORACLE
- F#, C# 3.0, LINQ Microsoft jet
- Scala twitter foursquare LinkedIn
- Haskell facebook BARCLAYS at&t
- Erlang facebook amazon T-Mobile
- OCaml facebook Bloomberg CITRIX

https://ocaml.org/learn/companies.html Jane Street

Functional languages help writing correct code

Abstraction:

Functional languages - mathematical functions
Imperative languages - mutable state

Imperative languages
Machines are good at complex manipulations of states, humans are not good at reason about them.

Functional languages
Variable never change values and functions have no side-effect, this makes it easier to reason about them.

A good functional language part of the ML family.

- Small core language,
- Supports first-class higher order functions,
- Lexically scoped
- Statically strongly typed
- Type inference
- It has a good community: ocalm.org

Informal Assignments

- Sign up or check access to **Piazza** and **Gradescope,**
- All communication with the teaching staff will be on **Piazza** – not via email.
- Get **OCaml** installed on your machines,
- Get **VS Code** (*Visual Studio Code*)installed on your machines.