

Welcome!

CS-214 - 9 Sept 2024
Clément Pit-Claudel

WELCOME TO THE SWISSTECH CONVENTION CENTER

THIS COURSE IS BEING HELD IN A PRIVATE BUILDING. ONLY USE THE ENTRANCE AND
FACILITIES CLEARLY INDICATED «COURS EPFL»

RESPECT THE SIGNS AND DON'T VISIT THE OTHER EVENTS SPACES

PLEASE DO NOT EAT OR DRINK IN THE AUDITORIUM

DON'T FORGET TO TAKE ALL YOUR BELONGINGS AND ANY WASTE WITH YOU

SHOULD YOU FORGET SOMETHING, NOTE THAT THE «LOST & FOUND» IS LOCATED AT THE
ENTRY OF THE AUDITORIUM

THANK YOU FOR YOUR COOPERATION !



Staff

3 profs

Martin Odersky (LAMP)

Viktor Kunčak (LARA)

Clément Pit-Claudel (SYSTEMF)

10 PhD assistants

Anna, Matt, Nguyễn, Samuel,
Sankalp, Shardul, Yann, Yaoyu,
Yawen, Yichen

14 BS / MS assistants

Arthur, François, Léa, Maël,
Thomas, Rémy, Aisel, Dylan,
Léo, Mai-Linh, Marwan,
Nguyễn, Sidonie, Valentin

Course platforms

Make sure you're
registered!

Course website

Syllabus
Policies
Useful links

Moodle

Links to materials
Lab submissions

Ed

All discussions
Supports private questions

MediaSpace (best effort)

Livestreaming
Course recordings

Schedule

First poll + demo lab out
“find” lab out Wednesday

Lectures

Mondays* (STCC)
Wednesdays (SG 1)

Exercises

Weekly (out Mon / Wed)

Polls

Released monthly

Labs

Weekly (out Wednesdays)
Due ~10 days later
Sometimes w/ a callback

Midterm

On paper, date TBD
Most similar to exercises

Final

On computers (likely), date TBD
Most similar to labs

Workload

EPFL: 8 credits $\approx$ 16 h/w

Our target: 9-15 h/w

Lectures

3 hours in class
+ 0-2 hours at home

Exercises (in groups)

~2 hours in help sessions
+ 1-2 hours at home

Labs (on your own)

~3 hours in help sessions
+ 0-3 hours at home

No additional project

Tips to be successful

Start with exercises

3h exercises + 4h lab

<< 0h exercises + 12h lab

Attend lectures

Ask questions

In person or on Ed

Don't LLM

Exams are offline 🙄

Work in groups

On exercises, not labs!

Plan ahead

Check the **syllabus**

Look out for deadlines



TODOS / DEADLINES

Set up Scala.
Read course policies.

Sep 11 (Wed)

→ Lab 1 ([find](#)) released.

Sep 13 (Fri), 23h

← Lab 0 ([setup](#)) due.

← Poll 0 ([Welcome](#)) due.

Sep 20 (Fri), 23h [Next week]

← Lab 1 ([find](#)) due.

NEW SYNTAX, APIS, AND TOOLS

Scala

`def =>... (by-name) {} if...then...else... val`

operators: `&&`, `||`, `==`, `<`, `>`, `<=`, `>=`, `+`, `*`, `-`, `/`, `%`

Basic types: `Int`, `Boolean`, `String`

But not: `var`, `for`, `while`

Tools

`SBT` (builds) `MUnit` (tests) `Git` (version control)

Command-line interfaces `Worksheets`

The Scala REPL

Plan ahead!

MONDAY

Intro to Functional Programming

Concepts

Functional programming Recursion

Evaluation strategies

Learning objectives

- Contrast functional and imperative styles.
- Write simple recursive functions in Scala.
- Reason about code execution using the substitution method.

WEDNESDAY

Version control

Concepts

Changesets / Commits / Revisions

History / Log Patches Blame

Learning objectives

- Describe the value of version control systems for individual & collaborative projects.
- Track and explore software history using Git.

LAB

Find

Concepts

APIs Tree traversals

Tests & Mocking

Goal

Implement a simpler version of the Unix `find` command (traverse a file system and print entries matching a rule).

MOST IMPORTANT EXERCISES

Scala [Horner polynomials](#) • [Set difference](#) • [Tree depth](#) • [String splitting](#)

Git [Creating patches](#) • [Browsing history](#)

CALLBACK

No callbacks until week 3!

“Have fun ! (No sarcasm)”

“Thinking about the exercises and labs on paper for a while before coding can really save a lot of time for some of them”

“Start doing labs in advance, not "one or two days before the deadline"”

“Have Fun and give the labs your all !”

Do not just try to get 100% on the labs, understand really what you do and why you do it.

“Be curious”

“I was too proud to ask for help (wanting the satisfaction of finishing a lab all by myself, without tips), and I didn't profit from learning to debug as much as I could have.”

“The code in scala doesn't do weird things, you did weird code.”

“Follow the debugging advices”

“Do the star exercices before starting the labs”

“Take advantage of the support provided in learning how to debug more. I regret not using the debugging guide as much as I should have.”

“See this class as different from the other programming classes you took, it is more about being a good programmer than about learning how to code in a language.”

Please answer the
welcome poll during
the break!