Spring 2023 CSC 330 A01 - A02 X

JJ

Search Topics

Bookmarks

Course Schedule 1

Table of Contents 6

Syllabus

OCaml: Week 1 ✓

OCaml: Week 2 ✓

Assignment 1 ✓

OCaml: Week 3 2

OCaml: Week 4 3

Assignment 2 1

Syllabus

This course teaches the basic concepts behind programming languages, with a strong emphasis on the techniques and benefits of functional programming.

Synopsis

- **Instructor**: Ibrahim Numanagić (/nu-maa-nagg-ich/) inumanag@uvic.ca
- **TAs**:
    - Zheng'an Zhao
    - Mazyar Ghezelji
- **Time & location**: TWF 10.30–11.20 @ CLE A224 (in-person)
- **Difficulty**: medium-hard
- **Office hours**:
        - Ibrahim: T 11.30–13.00 @ ECS 526
        - TAs: F 14.00–15.30 @ ECS 227

Communication

We will be using Brightspace (😩) for sharing course materials, and for submitting and grading assignments.

We will use Piazza for announcements, and to facilitate questions, answers and general class discussion. Please register here (the access code is `mahnahmanah`).

- Piazza is not hosted in Canada. If you are conscious of your privacy, feel free to use a pseudonym when registering. Piazza also supports anonymous comments, and you are free to use them.
- Note that we are using a free version of Piazza for this term. The free version will bug you to pay some amount of money in order to remove the annoying red banner. You do not have to do this! Just ignore the banner, or even better, find a way to block it—are you a computer science student or not?

The primary method of communication will be through Piazza. Students are free to help other students and collaboratively answer the questions.

- ⚠️ **DO NOT EMAIL TAs!** Use Piazza! They will NOT answer emails.
- ⚠️ **DO NOT EMAIL THE INSTRUCTOR!** Use Piazza unless you have a personal matter that must be discussed in private!
    - If you need to reach me privately, email me.
    - Assignment questions, project questions, or anything that is not related only to you is not a personal matter! Use Piazza!

Textbooks

No textbooks are needed. Supplementary materials will be posted on Brightspace when needed.

Grading

Your total grade (100%) will consist of the following components:

- [30%] **Assignments**:

  Six (6) assignments, each worth 5%, should be expected every two weeks. Assignments are to be done individually.

  - ⚠️ **IMPORTANT**: You will need to get at least one-third of the total assignment score to pass the course (i.e., at least 10% out of max. 30%)!
  - ➡️ **Note**: Some students might be chosen to describe one or several of their assignments in an oral interview.

- [60%] **Midterms**:

  Three (3) in-person midterms are tentatively scheduled on Feb 3, Mar 10, and Apr 5. The first two midterms are worth 22.5% of the total grade; the third midterm is worth 15% of the grade.

  - ⚠️ **IMPORTANT**: You will need to get at least one-third of the total midterm score to pass the course (i.e., at least 20% out of max. 60%)!

- [10%] **Final project** A final project, in the form of an extended assignment problem, will be due roughly two weeks after the last class.

Late policy

- Two late days per assignment; five in total for the whole semester (strict)
  - No help will be provided on Piazza during the late period
  - The same goes for weekends: if you submit "HELP ME!" on 23.50 Sunday... well, most likely you'll come short.

Learning outcomes

- You will learn the principles on which programming languages are built
- You should understand functional programming and become a capable functional programmer.
- You will learn a strongly-typed functional language (OCaml), a dynamically-typed functional language (Racket or Scheme), and either an object-oriented (Ruby) or an array-based programming language (J or Q).
- Ideally, you should become a better programmer.

Prerequisites

This course is not to learn to program. As a prerequisite to this course, **you are expected to be a *capable* programmer in one or two languages.**

You should have mastered at least one programming language, and know concepts such as recursion, expections, classes, objects, loops etc.

> If you cannot do FizzBuzz, please drop the course!

A high level of computer literacy is also expected. This includes installing new software, dealing with weird packages, using command-line tools, etc. Furthermore, familiarity with the Unix/Linux command line **is highly recommended**.

> **Note**: I am a UNIX user (Linux, macOS etc.) and not a Windows user. You are on your own when it comes to Windows setup issues. If you use Windows, save your souls and get WSL set up and running—you're welcome!

Course Outline

We will cover the following topics during the course of this semester:

- Syntax vs. semantics vs. idioms vs. libraries vs. tools
- Bindings, conditionals, records, functions
- Recursive functions and recursive types
- Algebraic data types, pattern matching
- Tail recursion
- Higher order functions, closures
- Scope
- Currying
- Type inference
- Abstract types and modules
- Dynamic vs. static typing
- Abstract types via dynamic type
- Pure object orientation
- Subtyping vs. parametric polymorphism
- Array programming (if type permits)

Some adjustments might be made over the course of the semester.

Inspiration

This offering closely follows the previous offering of this course by Daniel German, and CSE 341 by Dan Grossman at the University of Washington. Lots of useful stuff there!

Code of conduct

TL;DR:

- Be professional and respectful, and use common sense!
- Don’t cheat, for God’s sake!
    - It is OK to use some help. But make sure to acknowledge all sources properly:
        - Say: "Sami gave me an idea for the problem 3"
        - Say: "I found the idea on Wikipedia: here is the link ..."
        - Say: "I found this in a textbook XX by YY!"
        - but: **DON'T ADAPT OR COPY/PASTE OTHER PEOPLE'S CODE!**

        - If you get caught... heh, good luck
            - You will get **a zero** on the assignment (really! crying and wailing won't help at all)
            - Repeated offences will be met with F, and you might get expelled from the university
            - **I take these things seriously!**
- Don’t post offensive stuff on Piazza (c'mon, really...)
- If you have problems, let me know!
    - Use Piazza if you struggle with course materials— we can all help!
    - Any personal issue, as well as general course feedback, should be communicated to the instructor via the designated email.
- Do not share or distribute course material without my permission.

Full version...

`<legal-yada-yada>`

The university and the Faculty of Engineering has a strong mandate to support Equity, Diversity and Inclusion: https://www.uvic.ca/engineering/about/equity/index.php

We as a teaching team will do what we can to create a positive, safe, and supportive environment for you to participate in all components of this course offering.

You are expected to be respectful of other students and the instructor/TAs: when participating online, mute your microphone if you are not talking, participate by providing input, and ask questions using decent language and behaviour.

Strict monitoring of academic integrity will be performed in this course for any work submitted for marks. See course component descriptions and Course Policies and Guidelines below for details on academic integrity expectations. Substantiated academic integrity violations will be referred to the Department's Academic Integrity Committee which will determine penalty and ensure a record of the violation is kept with the university.

`</legal-yada-yada>`