

Schedule

A week-by-week breakdown of the material.

Week 1 (01/11-01/15)

Day 1 OCAML setup¹

Day 2 OCAML basics² Evaluation model, basic types, bindings

Day 3 Functions³

Day 4 Tuples, practice⁴

Assignment 1⁵ due by class time, Friday, January 22nd.

Week 2 (01/18-01/22)

Day 1 Lists and Option Types⁶

Day 2 Pattern-matching⁷

Day 3 Recursion⁸

Day 4 State recursion⁹

Assignment 2¹⁰ due by class time, Wed, January 27th.

Week 3 (01/25-01/29)

Day 1 Tail Calls¹¹

Day 2 Style reviews.

Day 3 Recursion practice.

Day 4 Assignment 2 style reviews. Recursion practice problems¹²

¹<notes/setup.html>

²notes/ocaml_basics.html

³notes/ocaml_functions.html

⁴notes/ocaml_functions.html

⁵<assignments/hw1.html>

⁶notes/lists_options.html

⁷notes/pattern_matching.html

⁸<notes/recursion.html>

⁹notes/recursion_state.html

¹⁰<assignments/hw2.html>

¹¹notes/tail_calls.html

¹²notes/recursion_state.html

Week 4 (02/01-02/05)

Day 1 Type aliases and Type variants¹³

Assignment 3¹⁴ due by class time, Monday, February 8th.

Day 2 Polymorphic Types¹⁵

Day 3 Type inference¹⁶

Day 4 Anonymous Functions, Functions as values¹⁷

Assignment 4¹⁸ due Thursday, February 11th.

Week 5 (02/08-02/12)

Day 1 Currying¹⁹

Day 2 Exceptions and exception handling²⁰

Assignment 5²¹ due Wednesday, February 17th.

Day 3 Sick day

Day 4 Sick day

Week 6 (02/15-02/19)

Day 1 Higher order functions²²

Assignment 6²³ due Monday, February 22nd.

Day 2 List functions²⁴

Day 3 Introduction to Modules²⁵

Assignment 7²⁶ due Friday, February 26th.

Day 4 References and mutation²⁷

¹³[notes/type_variants.html](#)

¹⁴[assignments/hw3.html](#)

¹⁵[notes/types_polymorphic.html](#)

¹⁶[notes/type_inference.html](#)

¹⁷[notes/functions_anonymous.html](#)

¹⁸[assignments/hw4.html](#)

¹⁹[notes/currying.html](#)

²⁰[notes/exceptions.html](#)

²¹[assignments/hw5.html](#)

²²[notes/functions_higher_order.html](#)

²³[assignments/hw6.html](#)

²⁴[notes/functions_list.html](#)

²⁵[notes/modules.html](#)

²⁶[assignments/hw7.html](#)

²⁷[notes/references.html](#)

Week 7 (02/22-02/26)

Day 1 Sick Day

Day 2 Delayed Evaluation²⁸

Day 3 Records and Objects²⁹

Day 4 Building an Interpreter³⁰

Assignment 8³¹ due Monday, March 14th.

Week 8 (02/29-03/04)

BREAK

Week 9 (03/07-03/11)

Day 1 Introduction to Racket³²

Day 2 **MIDTERM** (study guide³³)

Day 3 Introduction to Racket (cont)³⁴

Day 4 Bindings and Mutation in Racket³⁵

Week 10 (03/14-03/18)

Day 1 Macros³⁶

Dynamic Datatype-Programming via pairs³⁷

Assignment 9³⁸ due Monday, March 21st.

Day 2 Guest lecture

Day 3 Dynamic Datatype-Programming via structs³⁹

Day 4 Mutation in the Interpreter: The need for a store⁴⁰

²⁸[notes/delayed_eval.html](#)

²⁹[notes/records_objects.html](#)

³⁰[notes/interpreter.html](#)

³¹[assignments/hw8.html](#)

³²[notes/racket_intro.html](#)

³³[notes/midterm_study_guide.html](#)

³⁴[notes/racket_intro.html](#)

³⁵[notes/racket_bindings_mutation.html](#)

³⁶[notes/racket_macros.html](#)

³⁷[notes/racket_datatypes.html](#)

³⁸[assignments/hw9.html](#)

³⁹[notes/racket_datatypes.html](#)

⁴⁰[notes/interp_mutation.html](#)

Week 11 (03/21-03/25)

Day 1 Mutation in the Interpreter: The need for a store⁴¹
Assignment 10⁴² due Friday, April 1st.

Day 2 Mutation in the Interpreter: The need for a store⁴³

Day 3 Static Type-Checking vs Dynamic checking⁴⁴

Day 4 Interpreting objects and classes⁴⁵

Week 12 (03/28-04/01)

Day 1 Interpreting objects and classes (cont)⁴⁶

Day 2 Interpreting objects and classes (cont)⁴⁷

Day 3 Type-checking in the interpreter⁴⁸

Day 4 Type-checking in the interpreter⁴⁹

Week 13 (04/04-04/08)

Day 1 Memory Management and Garbage Collection⁵⁰

Day 2 Control Structures⁵¹

Day 3 Continuation-passing style⁵²

Day 4 Implementing Type Inference

Week 14 (04/11-04/15)

Day 1 TBA

Day 2 TBA

Day 3 TBA

Day 4 TBA

⁴¹[notes/interp_mutation.html](#)

⁴²[assignments/hw10.html](#)

⁴³[notes/interp_mutation.html](#)

⁴⁴[notes/static_vs_dynamic.html](#)

⁴⁵[notes/interpret_oop.html](#)

⁴⁶[notes/interpret_oop.html](#)

⁴⁷[notes/interpret_oop.html](#)

⁴⁸[notes/interpret_type_checking.html](#)

⁴⁹[notes/interpret_type_checking.html](#)

⁵⁰[notes/memory.html](#)

⁵¹[notes/control.html](#)

⁵²[notes/control.html](#)