Schedule

A week-by-week breakdown of the material.

Week 1 (01/11-01/15)

Day 1 OCAML setup[1]

Day 2 OCAML basics[2] Evaluation model, basic types, bindings

Day 3 Functions[3]

Day 4 Tuples, practice[4]

Assignment 1[5] due by class time, Friday, January 22nd.

Week 2 (01/18-01/22)

Day 1 Lists and Option Types[6]

Day 2 Pattern-matching[7]

Day 3 Recursion[8]

Day 4 State recursion[9]

Assignment 2[10] due by class time, Wed, January 27th.

Week 3 (01/25-01/29)

Day 1 Tail Calls[11]
Day 2 Style reviews.
Day 3 Recursion practice.
Day 4 Assignment 2 style reviews. Recursion practice problems[12]

[1] notes/setup.html
[2] notes/ocaml_basics.html
[3] notes/ocaml_functions.html
[4] notes/ocaml_functions.html
[5] assignments/hw1.html
[6] notes/lists_options.html
[7] notes/pattern_matching.html
[8] notes/recursion.html
[9] notes/recursion_state.html
[10] assignments/hw2.html
[11] notes/tail_calls.html
[12] notes/recursion_state.html

Week 4 (02/01-02/05)

Day 1 Type aliases and Type variants[13]

Assignment 3[14] due by class time, Monday, February 8th.

Day 2 Polymorphic Types[15]

Day 3 Type inference[16]

Day 4 Anonymous Functions, Functions as values[17]

Assignment 4[18] due Thursday, February 11th.

Week 5 (02/08-02/12)

Day 1 Currying[19]

Day 2 Exceptions and exception handling[20]

Assignment 5[21] due Wednesday, February 17th.

Day 3 Sick day

Day 4 Sick day

Week 6 (02/15-02/19)

Day 1 Higher order functions[22]

Assignment 6[23] due Monday, February 22nd.

Day 2 List functions[24]

Day 3 Introduction to Modules[25]

Assignment 7[26] due Friday, February 26th.

Day 4 References and mutation[27]

[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 Evaluation[28]

Day 3 Records and Objects[29]

Day 4 Building an Interpreter[30]

Assignment 8[31] due Monday, March 14th.

Week 8 (02/29-03/04)

BREAK

Week 9 (03/07-03/11)

Day 1 Introduction to Racket[32]
Day 2 **MIDTERM** (study guide[33])
Day 3 Introduction to Racket (cont)[34]
Day 4 Bindings and Mutation in Racket[35]

Week 10 (03/14-03/18)

Day 1 Macros[36]

Dynamic Datatype-Programming via pairs[37]

Assignment 9[38] due Monday, March 21st.

Day 2 Guest lecture

Day 3 Dynamic Datatype-Programming via structs[39]

Day 4 Mutation in the Interpreter: The need for a store[40]

[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 store[41]
Assignment 10[42] due Friday, April 1st.

Day 2 Mutation in the Interpreter: The need for a store[43]

Day 3 Static Type-Checking vs Dynamic checking[44]

Day 4 Interpreting objects and classes[45]

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 interpreter[48]
Day 4 Type-checking in the interpreter[49]

Week 13 (04/04-04/08)

Day 1 Memory Management and Garbage Collection[50]
Day 2 Control Structures[51]
Day 3 Continuation-passing style[52]
Day 4 Implementing Type Inference[53]

Week 14 (04/11-04/15)

Day 1 Implementing Type Inference (cont)[54]
Day 2 Implementing Parametric Types[55]
Day 3 Subtyping[56]

---

[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
[53] notes/interpret_inference.html
[54] notes/interpret_inference.html
[55] notes/interpret_inference.html
[56] notes/subtyping.html

Day 4 Subtyping (cont)[57]

[57] notes/subtyping.html