S0.4: Quiz 3 Preparation

CSci 2041:

Advanced Programming Principles

University of Minnesota,
Prof. Van Wyk,
Spring 2022

Logistics

- Wednesday, March 23
- It is an in-person quiz
- 9:05am, in our classroom (Anderson 310)
- 25 minutes allowed (for last 25 minutes of lecture)
- Bring you UCard and have your student ID# ready
- You may use 1 single-sided 8 1/2" by 11" sheet of handwritten notes.
  No other materials are allowed.
- Recall: lowest score out of the 5 is dropped
- Remaining time on Wednesday used for S4 material

Course material covered by the quiz

Slides

- S1.2: OCaml Basics
- S1.3: Higher Order Functions
- S1.4: Expressions, Values, and Evaluation
- S1.5: Inductive Types and Values (just the basics)
- S2.1: Modularity in OCaml
- S3: Testing

Other material

- Sample functions written in class for covered material. See the `Schedule of current and past lectures` page for list of sample programs used in each lecture.
- Techniques used in Hwks 01, 02, 03, Labs 02, 03, 04, 05, 06, 07.

Reading in textbook

- Chapter 1, Chapter 2, Chapter 4
- Chapter 5, Chapter 6, parts on testing

Description of course content covered by the quiz

Functional programming language features

- functions (recursive and non) over simple types such as integers and strings
- computations over lists and tuples, pattern matching of these values
- lambda-expressions, curried functions, higher order functions
- all inductive data types, lists, trees, options, etc.

Concepts

- parametric polymorphism,
- characteristics of strong static type systems
- difference between expressions and values, understanding of process of evaluation
- principles of testing, goals of testing

Description of course content covered by the quiz

Programming techniques and patterns from in-class functions and Hwk 01

- arithmetic computations
- functions searching a range of values, *e.g.* `is_square`
- list processing functions:
  - operating on all values (*e.g.* increment all),
  - selecting certain values (*e.g.* find all evens),
  - selecting a single value (*e.g.* find longest string),
  - combining values in a list (*e.g.* sum all integers)
- using higher order functions, such as `map`, `filter`, `fold_left`, and `fold_right` over lists
- recursive and higher-order functions over inductive data types
  - deriving a reduce function from an inductive data type
- uses (not specifics) of OCaml modules, signatures, and functors

Format

- Short answer questions - 2 or 3 sentence answers
- Write small functions, syntax needs only be close to OCaml
- Read small functions, explain behavior

Material not covered by the quiz

Slides:

- S0.*: Course Introduction and Quiz Preparation slides
- S1.1: Whirlwind Tour
- S4: Reasoning About Correctness

Reading in textbook

- Any section not listed above as being covered