

Study Guide for Midterm Exam 1

Topics

The midterm exam 1 will cover the following topics and skills:

Language Classification (Week 1) [~8 pts]:

- 1) Distinguish between declarative languages and imperative languages
- 2) Describe characteristics of the von Neumann subfamily of imperative languages and the relationship between imperative languages and the von Neumann computer architecture
- 3) Describe characteristics of the scripting subfamily of imperative languages.
- 4) Describe characteristics of the functional subfamily of declarative languages, and give examples of functional languages.
- 5) Give examples of languages that can be both functional and imperative.

Compilation vs Interpretation (Week 2) [~12pts]:

- 1) Describe the model of pure compilation and pure interpretation
- 2) Compare advantages/disadvantages of compilation and interpretation
- 3) What is the difference between a compiler and a preprocessor?
- 4) Describe the following implementation strategies:
 - a. Library of Routines and Linking
 - b. Post-compilation Assembly
 - c. The C Preprocessor
 - d. Source-to-Source Translation (C++)
 - e. Bootstrapping, with example in Pascal

Functional Programming Features (Week 3) [~15pts]:

- 1) Describe classification of values/objects and its examples
- 2) Describe how strings in Java satisfy criteria of first-class objects
- 3) Which criteria of first-class objects are not satisfied by arrays in C++? Explain why not.
- 4) Which languages support first class functions?
- 5) Describe how to use lambda expressions in Java (and how to use interfaces and classes in Java)
- 6) What are lambda expressions in lambda calculus? Write lambda expressions in lambda calculus that describe identity function, constant functions, arithmetic functions, and function application.
- 7) What are necessary and sufficient conditions for a function to be called computable, based on Church-Turing thesis?

- 8) What is a higher-order function? Describe common uses (including currying) of higher-order functions. Explain why higher-order functions aren't common in imperative languages?

Scope and Binding (Week 4) [~13 pts]:

- 1) What is binding time? What is the static binding? What is dynamic binding?
- 2) Describe relationship between binding lifetime and object lifetime.
- 3) Describe storage allocation mechanisms for objects. Give examples of static objects, stack objects, and heap objects.
- 4) Describe static scope rules. Explain the effect of static scope rules in the output of a program.
- 5) Describe dynamic scope rules. How and where dynamic scope rules are used?
- 6) Describe how scoping is implemented.

Racket [~12pts]: In this exam, you are expected to understand and apply the syntax of Racket for

- 1) Simple values: numbers, strings, booleans
- 2) Defining functions
- 3) Conditionals: `if`, `and`, `or`, `cond`
- 4) lambda expressions
- 5) local bindings using `let` and `let*` form
- 6) Pre-defined functions: `strings?`, `string-prefix?`, `string-suffix?`

Other Guidelines

In this exam, you can use **one** cheat sheet of the standard US letter size. You can write on both sides of the cheat sheet.

Most questions are multiple-choice or true/false. There will be 2-3 questions that ask you write Racket code. Please make sure to bring pencils to the exam.

To study for exam, you should

- review the lecture presentation handouts of **weeks 1-4**
- read the related sections in the textbook and the Racket guide
- review homework assignments of **weeks 2-4**.