

## COMP 141: Midterm Exam Study Guide

The exam material includes:

- Course modules:
  - Introduction,
  - Language design Criteria
  - Syntax
  - Functional programming: Haskell (Section 1)
- Lecture material (slides, lecture videos, and labs) for the above modules.
- Selected material from textbook chapters 1, 2, and 6 (indicated below), and “Learn You a Haskell”.

The exam will be **closed-book**. You may use a basic **calculator** during the exam (you won’t need it!).

You will be expected to write and interpret small programs in Haskell. While language syntax is not as important as semantics, the syntax of Haskell is comparatively minimal, and you should be able to write code with reasonably correct syntax. You can use GHC/GHCi during the exam, i.e., you can verify your Haskell answers using online or locally installed Haskell compiler.

The following highlights ideas that are important for the examination. Omission of a topic from these tables does not mean that the topic will not appear on the exam. Explicit exclusion (in *italics*) of a subsection or topic does mean that that content will not be on the exam.

### 1. Module: Introduction

- The origins of PLs, computational paradigms, PL definition, PL translation, and the future of PLs.
- Slides, lecture videos and labs of this module
- Textbook Chapter 1
- *Material in textbook section 1.2 will not be covered.*

### 2. Module: Language Design Criteria

- Successful PLs and different PL design criteria
- Slides, lecture videos and labs of this module
- Textbook Chapter 2
- *Material in textbook sections 2.6 and 2.7 will not be covered.*

### 3. Module: Syntax

- Lexics, CFGs, Parse trees, ASTs, ambiguity, associativity and precedence, EBNFs
- Slides, lecture videos and labs of this module
- Textbook Chapter 6
- *Material in textbook sections 6.6, 6.7 and 6.8 will not be covered.*
- *Syntax diagrams in section 6.5 will not be covered.*

### 4. Module: Functional PLs: Haskell (Section 1)

- Programs as functions, introduction to FPLs, arithmetic expressions, boolean algebra, comparison operators, function definition, if expressions, name binding, lists, built-in functions on lists, laziness vs. eagerness in evaluation, list ranges, list comprehension, tuples, built-in types, built-in type classes, pattern matching
- Slides, lecture videos and labs of this module
- “Learn You a Haskell” Chapters 1,2,3: Full
- “Learn You a Haskell” Chapters 4: Up to “Guards”.