Book Assignment #1: Introduction

Issued: Tuesday, September 10
Due: Tuesday, October 1

Purpose

This assignment asks you to begin thinking about programming languages and programming-language translation. For each exercise, read it in the textbook, then adjust it according to what's here.

  1.1 This question asks you to give examples of various kinds of errors. Use Java for your examples.

  1.6 This question asks you to consider how a parse/syntax tree is traversed differently, during interpretation versus compilation. In our upcoming Translator Assignment, our translator performs interpretation *and* compilation.

  1.8 This question asks you to consider the accuracy of `make`. We discussed `make` during the lecture on a typical Unix C/C++ toolchain. Note that plain-old `make` relies on file-system time-of-last-modification timestamps.

  2.1 Only do parts a, b, and c.

   This question asks you to construct regular expressions for strings in three regular languages: character-string literals in C, comments in Pascal, and numeric literals in C.

   Specify the simplified regular languages as described in our textbook, not the real ones from real C and Pascal!

   For each of the three parts, start by giving some example strings in the regular language.

 2.13 Only do parts a and b.

   This question asks you to give a parse tree and a rightmost derivation, of a string, according to a context-free grammar.

Note that we are not covering material past Section 2.1, including scanning algorithms, parsing algorithms, and grammar characteristics (e.g., LL(1)).

2.17 This question asks you to extend the grammar from Figure 2.25.

Note that we are not covering material past Section 2.1, but we can still use the grammar as an example.

In our upcoming Translator Assignment, you will extend a provided grammar, scanner, and parser, in a similar way.