ASSIGNMENT 0 — COMPUTER LANGUAGES

COMP 3010 — ORGANIZATION OF PROGRAMMING LANGUAGES

1. Compilers and Interpreters

Exercise 1 (Short Answer)**.** Explain the task of each of the following phases of a compiler:

(1) Lexical analyzer (*a.k.a.* lexer)
(2) Syntax analyzer (*a.k.a.* parser)
(3) Semantic analyzer
(4) Intermediate code generator
(5) Code optimizer
(6) Machine code generator

Exercise 2 (Short Answer)**.** Explain the difference between a compiler and an interpreter.

Exercise 3 (This or That)**.** Which of the following are advantages of compilers? Which are advantages of interpreters?

(1) Can pre-examine input program for semantic (*e.g.,* type) errors
(2) Have full knowledge of both program input and program implementation
(3) Can specialize parts of code to optimize for specific input data
(4) Can afford heavy weight optimizations over large sections of code
(5) Flexible, and can easily change program behavior dynamically at run-time
(6) Generated code can run many times

2. Computability

Exercise 4 (Short Answer)**.** (1) If Q is a program that solves the halting problem, what must $Q(P, x)$ do?
(2) What is the significance of the halting problem for computable functions?

Exercise 5 (Short Answer)**.** Suppose that the program Z is a *zero checker*. That is, Z takes two inputs, a string P representing a program and a number n, and checks whether or not running P with input n (written as $P(n)$) would output a 0:

$$Z(P, n) = \begin{cases} \texttt{true} & \text{if } P(n) \text{ returns } \texttt{0} \\ \texttt{false} & \text{otherwise} \end{cases}$$

Because the halting problem is uncomputable, we know that the *zero checker* is not a computable function. Show why the program Z cannot exist by explaining how it could be used to solve the halting problem.

Date: Fall 2025.