# Compiler Construction/Spring 2020
# Homework 1

### Reading Assignments

- Lecture 1: Dragon Book §§1.1–1.2 (12 pages).
- Lecture 2: Dragon Book §§3.1–3.4, §§3.6–3.7 (50 pages).

### Homework Assignments

## 1 Compiler Architecture

**Question 1.1** (3 points)**.** Compilers typically consist of a frontend (lexing, parsing, semantic analysis, IR generation) and a backend (code generation). What advantages does this architectural separation bring?

**Question 1.2** (3 points)**.** Many compilers have multiple optimizers, at different stages in the pipeline. Why is it useful to have multiple optimizers?

**Question 1.3** (3 points)**.** Some compilers use multiple IRs. Find at least two examples and briefly explain the advantages it brings.

**Question 1.4** (3 points)**.** Some compilers translate a high-level language into another high-level language rather than assembly or machine code. What particular language is a common target for such compilers and why?

## 2 Compiler Phases

These questions are based on Figure 1.7 on page 7 of the Dragon Book. Your task is to determine the output of each of the compiler phases for the following source code statement:

```
celsius = (fahrenheit - 32) * (5/9).
```

**Question 2.1** (3 points)**.** What sequence of tokens does the lexical analyzer output?

**Question 2.2** (3 points)**.** What abstract syntax tree does the syntax analyzer output?

**Question 2.3** (3 points)**.** What is the abstract syntax tree after the semantic analyzer modifies it?

**Question 2.4** (3 points)**.** What sequence of three-address instructions does the IR generator output?

**Question 2.5** (3 points)**.** What sequence of three-address instructions does the optimizer output?

**Question 2.6** (3 points)**.** What sequence of machine instructions does the code generator output? Use the same assembly language as the Dragon Book or improvise your own.