## B1. Compilers (25 points)

1. (15 points) Consider the grammar below where E and G are nonterminals, &, #, (, ), and 'a' are all terminals:

```
E -> E & E | E # E | G
G -> a ( E ) | a
```

a. Describe all the reasons that the grammar is not suitable for producing a predictive, top-down parser.

b. Rewrite the grammar to enable a predictive, top-down parser construction.

c. For the *original* grammar above, show the sets of LR(1) items and the transitions of the canonical LR(1) DFA. Justify whether or not the grammar is LR(1).

2. (10 points)
What are the main operations for a symbol table? Discuss the data structures associated with a symbol table maintained as a list of hash tables and how the operations of a symbol table are implemented in that case. Give an example of what your symbol table would look like for a sample program.

**B2. Compilers (25 points)**

1. (15 points)
   a. Describe the process taken by a straightforward scanner generator. Be sure to describe the steps as well as the data structures that are typically used by a scanner generator. What are the two common ways to implement a scanner?

   b. State 2 features of a programming language that make it difficult to scan. Explain why these features make it difficult to scan, and how these features can be handled by the compiler.

   c. Write a regular expression for comments, consisting of a string of characters surrounded by /* and */, without an intervening */, unless it is inside double quotes (").

2. (10 points) In the following program, there are six temporaries a, b, c, d, e, f. What is the fewest number of registers that is needed for this program, without spilling? Justify your answer by showing the interference graph and a coloring of the interference graph. Show also the program after register allocation.

```
a = 1
b = 10
c = a + b
d = b + c
e = a + c
f = b + d
d = c + f
d = d + f
b = d + e
return d + f
```

## B3. Compilers (25 points)

1. (10 points) Static and dynamic storage allocation:

   a. Describe a language feature that would require static storage allocation and one that would require heap storage allocation. Explain in detail why this feature would require the particular type of storage allocation.

   b. Describe an additional language feature that would not be appropriate for static storage allocation and one that would not be appropriate for heap allocation. Do not use the same language features as in A. Explain in detail why these features would not be appropriate for the particular type of storage allocation.

2. (4 points) Describe the major differences and compare advantages/disadvantages between implementing a language via building a **compiler, interpreter**, or a just-in-time (**JIT**) compiler.

3. (4 points) Give an example of parametric polymorphism and explain how it complicates type checking.

4. (3 points) Describe the extra overhead of a dynamic dispatch versus a static dispatch.

5. (4 points) Show examples to demonstrate the distinction between overloading and overriding. How is code generated for the correct function calls with overloaded functions versus over-ridden functions?

**B4. Compilers (25 points)**

1. (15 points) Code Generation:

a. Give an algorithm to translate straight-line stack machine code into code for the RISC machine, such that each stack machine instruction is translated into a fixed pattern of RISC instructions; precisely describe all such patterns. The RISC code will ***emulate*** the operand stack in memory, and use a dedicated register $r_{SP}$ to point to the top of the emulated stack. Define how each of the stack machine instructions is translated into RISC instructions.

b. Give an algorithm to translate straight-line stack machine code into efficient code for the RISC machine. The code must ***not emulate*** the operand stack in memory; instead, registers of the RISC machine must be used, and they must be used as efficiently as possible.

2. (10 points)

a. Justify the names for the three well known garbage collection techniques: reference counting, mark-and-sweep, and copying collectors. Describe how the names indicate how garbage is identified and collected.

b. Choose two of these collectors and discuss the tradeoffs between them.