Instructions: This is an open book, open note, 110 minute exam. There are 3 questions worth a total of 100 points. You must answer all of the questions. Put your name on all pieces of paper that you turn in. Show all of your work, write legibly and good luck.

1) (70 points) You are to add a new language feature to ESJ and enhance your compiler appropriately. The language currently supports integer and boolean scalar types, you are to add a `float` type for floating point numbers.

The following code fragment illustrates some simple floating point code that your enhancement should be able to handle:

```
int x;
float y;
x = 2;
y = 1.5;
if (y > 0)
  y = y * x;
```

`float`s should work like any other builtin scalar type except that you only need to worry about the following operators: `+`, `-`, `*`, `/`, and the relational operators (including `==`).

Integers should be automatically promoted to floats when they appear in an expression with a float operand, e.g., the multiplication expression shown above. The JVM has a bytecode for promoting integers called "i2f" which pops, converts, and pushes the result.

The arithmetic operators, constant loading, stack load/store, and return bytecodes are the same as for integers except they have a leading "f", e.g., "fadd", "fconst", "fload", "freturn".

Unfortunately, none of the nice conditional branch instructions, such as "if_icmpeq", are available for floats. Instead there are two bytecodes "fcmpl" and "fcmpg" which perform less-than and greater-than comparisons respectively on the top two float values on the stack and push a boolean result on the stack.

Make any reasonable assumptions that you think are necessary, e.g., a float fits in a single stack slot, but, be sure to explicitly state your assumptions.

1 continued) In answering the following questions, I expect you to describe in detail the changes you would make to the ESJ compiler. You may use a mixture of descriptions including English text, compiler notations (e.g., regular expressions), diagrams, and pseudo-code sketches. I do not expect you to write down correct Java code.

a) **(10 pts)** Enhance the scanner to handle the new lexical elements.

b) **(10 pts)** Enhance the parser to handle the new syntactic elements. Illustrate the parse tree for the example given above.

b) **(10 pts)** Enhance the AST construction actions in the parser to handle the new languge elements. Illustrate the AST for the example given above.

c) **(20 pts)** Sketch out type rules for floats including rules that handle type coercions; focus on expressing the essential ideas of the type rules. If multiple aspects of the float language extension are very similar, you need only describe one of them and briefly explain how it would change for the other aspects. Illustrate the type derivation tree for the example given above and indicate whether it is type correct.

e) **(20 pts)** Enhance the code generator to handle `float`s. Be sure to show code generation templates for all new constructs. Illustrate the generated JVM bytecodes for the given example.

2) (16 points) This question is about how to use the results of static analysis in optimizing code generation. You have studied *live variables* (LV) analysis. In this problem, you are to describe two different approaches that use the results of LV analysis to improve the code generated by the compiler. You may refer to bytecode generation, native code generation, peephole optimization, or any other approach to generating or improving code.

Each approach is worth **(8 pts)** in total. For full points, you must

- explain how the information in LV is used;
- explain how the code is improved;
- explain why the improvement is correct; and
- illustrate the improvement on a small example.

3) (14 points) This question is about compiler architecture and the structure of the ESJ compiler in particular.

a) **(4 pts)** Define *compiler phase* and *compiler pass*.

b) **(10 pts)** Identify the phases and at least 5 passes in the ESJ project compiler. Of those 5 passes, at most 2 of them can come from the same phase. Give a one sentence description of what each of the phases and passes does.