**Name:** _____

1. (1 pt.)
   - **Read all material carefully.**
   - *If in doubt whether something is allowed, ask, don't assume.*
   - You may refer to your **books, papers, and notes** during this test.
   - **E-books** may be used *subject to the restrictions* noted in class.
   - **Computers** (including smart phones, tablets, etc.) **are not permitted**, except when used strictly as e-books or for viewing ones own notes.
   - **Network access** of any kind (cell, voice, text, data, ...) is **not permitted**.
   - Write, and draw, carefully. **Ambiguous or cryptic answers receive zero credit.**
   - Use **class and textbook conventions** for notation, algorithmic options, etc.
   - **Do not attach or remove any pages.**

**Write your name** in the space provided above.
**Do not write anything else on this page.**

WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.

(Do not view any other pages.)

**Do not write on this page.**
(It is for use in grading only.)

| Q | Full | Score |
|---|---|---|
| 1 | 1 | |
| 2 | 9 | |
| 3 | 15 | |
| 4 | 20 | |
| total | 45 | |

2. (9 pts.) Provide a (1) **leftmost derivation**, (2) **parse tree**, and (3) **abstract syntax tree** for the input (sentence) `f o o h e e s` using the following grammar (using *yacc/PLY* syntax):

```
S : F H X
F : f | f F
H : h | o H e
X : s | s F s
```

3. (15 pts.) Provide a complete JCoCo assembly language program that reads two whitespace-separated integers, $i$ and $j$ from a single line of *standard input* and prints the value of $i^2 + j$ on standard output. *Explain why your program is correct.*

[additional space for earlier material]

4. (20 pts.) Consider the *JCoCo* virtual machine running the following JCoCo assembly language program. The `# *` tags are comments.

   (1) Depict the state of the **operand stack** after each instruction that is marked with the comment `# *`.

   (2) Depict the state of the **call stack** at least once, when it contains more than one stack frame.

   (3) State any output produced by the program.

   *Provide brief explanations to qualify for better partial credit.* Reminder: **Use of computers to is not permitted. Running the program using *coco* or similar is not allowed.**

```
Function: f101/2
Locals: x, y
BEGIN
          LOAD_FAST                     0
          LOAD_FAST                     0          # *
          BINARY_MULTIPLY
          LOAD_FAST                     1          # *
          BINARY_ADD
          RETURN_VALUE
END
Function: main/0
Constants: "V", 4, 13, "I'm", "Jason"
Globals: print, f101
BEGIN
          LOAD_CONST                    3
          LOAD_CONST                    4
          LOAD_CONST                    0         # *
          BUILD_LIST                    3         # *
          GET_ITER                                # *
label00:  FOR_ITER                label01         # *
          LOAD_GLOBAL                   0
          ROT_TWO                                 # *
          CALL_FUNCTION                 1
          POP_TOP
          JUMP_ABSOLUTE           label00
label01:  LOAD_GLOBAL                   0
          LOAD_GLOBAL                   1
          LOAD_CONST                    1
          LOAD_CONST                    2         # *
          CALL_FUNCTION                 2
          CALL_FUNCTION                 1
          RETURN_VALUE
END
```

[additional space for earlier material]

[additional space for earlier material]