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.

   Write your name in the space provided above.

WAIT UNTIL INSTRUCTED TO CONTINUE TO REMAINING QUESTIONS.

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

Q	Full	Score
1	1	
2	19	
3	15	
4	15	
5	15	
total	65	

2. (19 pts.) Consider the *JCoCo* virtual machine running the following JCoCo assembly language program. Depict the state of the operand stack after each instruction. State any output produced by the program. *Provide brief explanations to qualify for better partial credit.* Reminder: Use of computers is not permitted. Running the program using *coco* or similar is not allowed.

```
Function: main/0
Constants: 2, 3, 7, None
Locals: my, name, is
Globals: print
BEGIN
  LOAD_CONST  0
  STORE_FAST  0
  LOAD_FAST  0
  LOAD_FAST  0
  LOAD_FAST  0
  BINARY_MULTIPLY
  STORE_FAST  1
  LOAD_FAST  1
  LOAD_FAST  1
  LOAD_CONST  1
  BINARY_MODULO
  LOAD_CONST  2
  BINARY_SUBTRACT
  STORE_FAST  2
  LOAD_FAST  2
  LOAD_FAST  0
  LOAD_FAST  1
  BINARY_TRUE_DIVIDE
  LOAD_FAST  2
  BINARY_MULTIPLY
  LOAD_GLOBAL  0
  ROT_TWO
  CALL_FUNCTION  1
  POP_TOP
  LOAD_CONST  3
  RETURN_VALUE
END
```

[additional space for earlier material]

3. (15 pts.) Provide a complete JCoCo assembly language program that reads an integer n from *standard input* and prints the value of n^4 (the input number raised to the fourth power) on standard output. *Explain why your program is correct.*

4. (15 pts.) Provide a context-free grammar (CFG) for parsing lists in the format used by Homework HW01. The grammar should be designed to accept a single list (of arbitrary length) in that format, and nothing else. Be sure to use exactly the format specified in the homework. *Explain why your CFG is correct* (why it accepts all valid lists and also why it accepts nothing else).

5. (15 pts.) Provide (1) **leftmost derivations**, (2) **parse trees**, and (3) **abstract syntax trees** for the following input (sentence) using the grammar of Question 4:

```
(3, 1, 4)
```

[additional space for earlier material]