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.

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	9	
3	15	
4	20	
total	45	

2. (9 pts.)

   (a) Which of *static scope* and *dynamic scope* does standard Python use?

   (b) What is the output of the Python program below? **Explain your answer** using concepts such as binding of free variables, nesting of code, state of the run-time stack, etc.

   (c) What would the output be if, instead of the option chosen in your answer to Question 2a above, Python used the other option? **Explain your answer** using concepts such as binding of free variables, nesting of code, state of the run-time stack, etc.

```python
greeting = "Howdy"
def makeGreeter(greeting, name):
    def greeter(name):
        return greeting + ", " + name + "!"
    return greeter
print(makeGreeter("Aloha", "Alice")("Bob"))
```

3. (15 pts.) Consider the JCoCo assembly language program listed below.

   (a) (5 pts.) **Explain** what the program does as precisely as possible. In particular, describe its output as a function of its input.

   (b) (10 pts.) Provide a **complete JCoCo assembly language program** that exhibits the same input-output behavior as this one but whose code is shorter by at least one instruction, or explain why no such shorter program is possible. If your shorter program reuses parts of this program then you may indicate so instead of rewriting those parts **but only if** the result is completely obvious and unambiguous. **Explain why your answer is correct.**

```
Function: main/0
Constants: None, "Hello, ", "!"
Locals: myname, greeting
Globals: input, print
BEGIN
          LOAD_GLOBAL                    0
          CALL_FUNCTION                  0
          STORE_FAST                     0
          LOAD_CONST                     1
          LOAD_FAST                      0
          BINARY_ADD
          LOAD_CONST                     2
          BINARY_ADD
          STORE_FAST                     1
          LOAD_GLOBAL                    1
          LOAD_FAST                      1
          CALL_FUNCTION                  1
          POP_TOP
          LOAD_CONST                     0
          RETURN_VALUE
END
```

[additional space for earlier material]

4. (20 pts.) Provide a **complete JCoCo assembly language program** that

   (a) Reads two newline-terminated strings from *standard input.*

   (b) Writes the concatenation of those two strings (in input order), separated by a space, to *standard output.*

   (c) **Explain why your program is correct.**

[additional space for earlier material]

[additional space for earlier material]