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. (No sharing.)
   - **E-books** may be used **subject to the restrictions** noted in class. (Briefly, do only those things with an e-book that are just as easily done with a physical book.)
   - **Computers of any kind** (including tablets, phones, and similar devices) are **not permitted** except when used exclusively as e-book readers.
   - **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.
   - Questions that ask for **explanations** allocate a sizable fraction of points to those. **Answers without sufficient explanations will score very poorly.**
   - Do not add, remove, detach, or mangle pages (causes scanner problems).
   - Budget your **time**, noting that *number of points = number of minutes*.

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	14	
3	15	
4	15	
total	45	

2. (14 pts.) This question centers on the issue of *scope* as discussed in class and the textbook. What is the output of the following Python program? **Explain your answer using concepts related to scope.**

```
def main():
    x = 10
    y = 20

    def f(x):
        def g():
            return x + y

        return g

    y = 30
    print(f(3)())


main()
```

3. (15 pts.) For each of the following *Standard ML* expressions, provide the response when that expression is evaluated by the `sml` REPL (read-eval-print loop). Assume that the expressions are evaluated in the order listed. In your response, *draw a* ***box*** *around the* ***type*** *and* ***`oval`*** *around the* ***value****.* (If there is an error then clearly explain the error.) **Briefly explain how the types are inferred.**

   (a) (5 pts.) `fun f401 (x) = x + 42;`

   (b) (5 pts.) `fun f402 (x, y) = x / y;`

   (c) (5 pts.) `fun f403 (x, y) = 1 :: x :: y;`

4. (15 pts.) Provide a **complete JCoCo assembly language program** that

   - Reads two newline-terminated strings from *standard input* (one per line).
   - Writes a single integer n followed by a newline to *standard output*, where n is the product of the lengths of the two strings read above.

   Hint: The Python function `len` is available in JCoCo.
   Explain why your program is correct.

[additional space for earlier material]