

Name: _____

1. (1 pt.)

- **Read all material carefully.**
- This test is **closed book, closed notes**.
- However, you may refer to **one** standard Letter-sized sheet of paper (both sides) that has **notes hand-written by you**. If used, these notes must **include your name** near the top and must be **submitted** along with this test.
- Computing or communication devices of any kind (laptop computers, tablets, phones, calculators, etc.) are not permitted.
- Network access of any kind (cell, voice, text, data, etc.) 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 only in designated areas of this test. Do not separate the pages and do not include any additional paper.

Print your name clearly 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	13
3	10
4	15
5	6
total	45

2. (13 pts.) Consider the JCoCo assembly language program listed below. **Explain** what the program does *as precisely as possible*. (Recall recent classroom discussion of similar questions.) In particular, describe its *output as a function of its input*. Provide some *illustrative sample* input and the corresponding output.

```

Function: main/0
Constants: None, "", " : "
Locals: s, t
Globals: input, print
BEGIN
    LOAD_GLOBAL          0
    LOAD_CONST           1
    CALL_FUNCTION        1
    STORE_FAST          0
    LOAD_GLOBAL          0
    LOAD_CONST           1
    CALL_FUNCTION        1
    STORE_FAST          1
    LOAD_GLOBAL          1
    LOAD_FAST            1
    LOAD_CONST           2
    BINARY_ADD
    LOAD_FAST            0
    BINARY_ADD
    CALL_FUNCTION        1
    POP_TOP
    LOAD_CONST           0
    RETURN_VALUE
END

```

[additional space for earlier material]

3. (10 pts.) Provide a **complete JCoCo assembly language program** that exhibits the same input-output behavior as the program of Question 2 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.**

4. (15 pts.) Provide a **complete JCoCo assembly language program** that
- (a) Reads a single newline-terminated string from *standard input*.
 - (b) Writes a single line of text to *standard output*. That text is **short** if the length of the input string is strictly less than 10 and **long** otherwise.

Provide a detailed explanation of the code using a suitable combination of comments adjacent to the code and a separate description that together make its design, working, and correctness very easy to understand. (Inadequate or missing explanations will lead to a score very close to zero.)

[additional space for earlier material]

5. (6 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.

(a) (2 pts.) `42.42 + 2.0;`

(b) (2 pts.) `42.42 + 2;`

(c) (2 pts.) `fn x => x + 101;`