UMassAmherst | Manning College of Information & Computer Sciences

Programming Methodology

# Lab 4: Closures and Mental Models

Wednesday February 25, 2026

# Weekly Lab Agenda

- Go over reminders/goals
- Review past material
- Work in groups of 2-3 to solve a few exercises.
  - Please sit with your group from last week.
- Discussion leaders will walk around and answer questions.
- Solutions to exercises will be reviewed as a class.
- Attendance will be taken at the end.

# Reminders

- Midterm 1 is next Wednesday (3/4) from 7-9 pm.
  - Previous exams can be found under “Modules” on Canvas.
- Homework 4 is due tonight (2/25) at 11:59pm.
  - Come to office hours for help!
- Homework 5 will be released after the midterm.
- If you need to miss lab and have a valid reason according to the syllabus (medical, other personal) please fill out the lab excusal form on Canvas before the start time of your lab.
  - Waking up late or the bus being late are NOT valid reasons to miss lab.

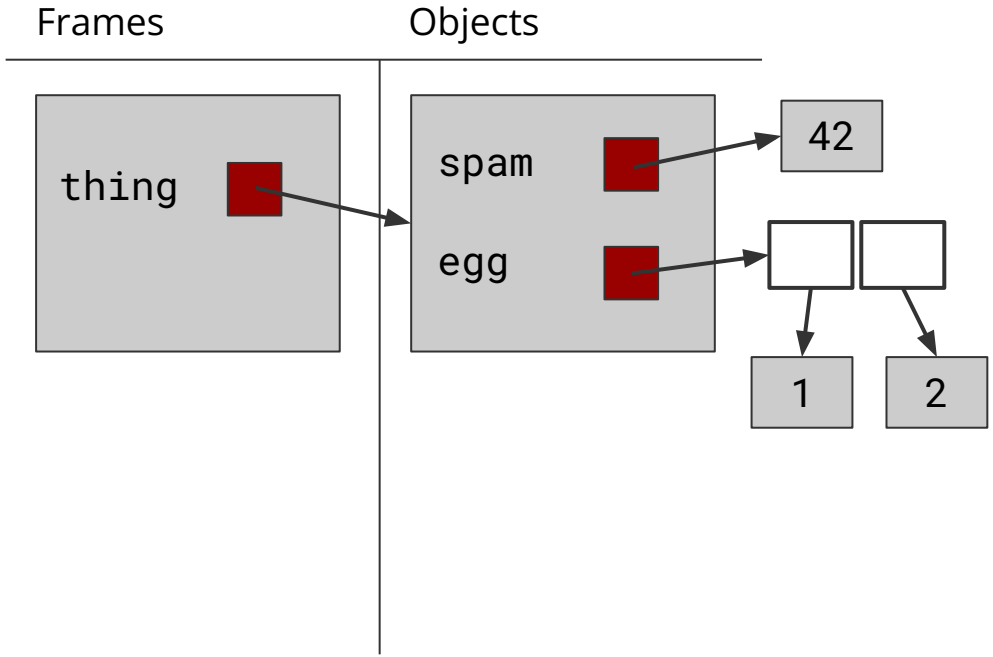
# Today's Goals

- Practice creating mental models
- Practice test design and using Jest

# Exercise 1: Mental Models

The memory diagram for the following code is depicted on the right. It uses reference semantics for all types.

```
1 let thing = {
2     spam: 42,
3     egg: [1, 2]
4 };
```



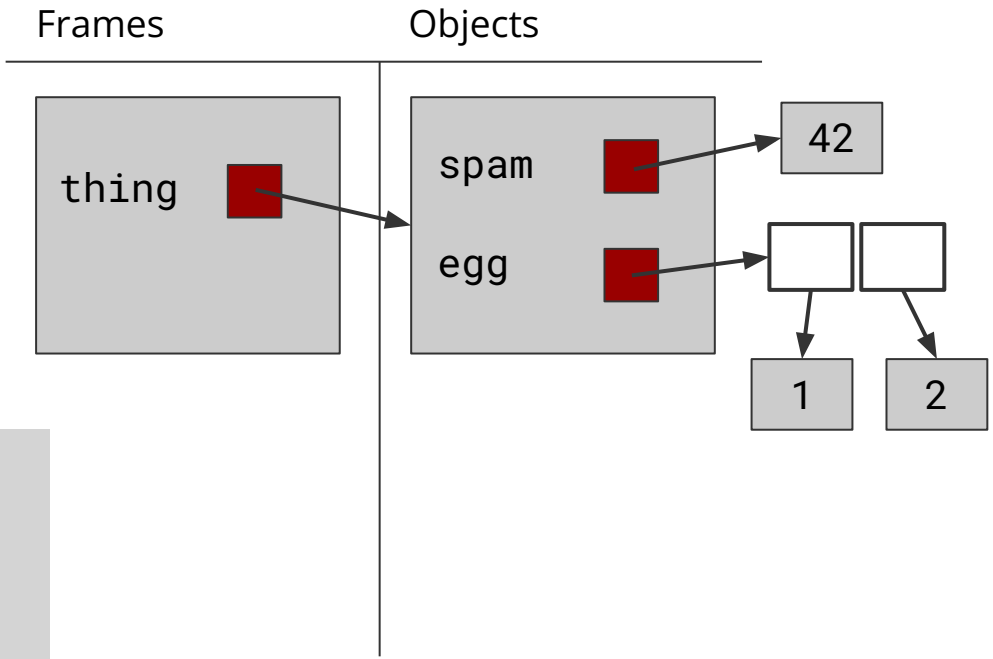
# Exercise 1: Mental Models

```
1 let thing = {
2     spam: 42,
3     egg: [1, 2]
4 };
```

Draw the updated memory diagram after running lines 5-10 below.

**Note**: "a += b" is equivalent to "a = a + b" in TypeScript.

```
5  let tmp = thing.egg;
6  thing.egg[0] += 3;
7  thing.egg = thing.spam;
8  thing.spam = tmp;
9  thing.spam.push(thing.egg);
10 thing.spam.filter(x => x < 5);
```



# Review : Jest

**Jest** is the **testing framework** that we use in this course.

```
// each function has a ‘describe’ block
describe(“func”, () => {
    // a ‘describe’ block can have multiple ‘it’ blocks
    // each ‘it’ block represents a test case for the function
    it(“should do something”, () => {
        ...
    });
    ...
});
```

# Review : Jest

In addition to global functions, Jest provides robust testing utilities like `expect`.

```
describe("func", () => {
    it("should do something", () => {
        ...
        // expect can be chained with 'matchers'
        expect(a).toEqual(b);
        expect(f).toThrow(e);
    }); // see jestjs.io/docs/expect for more
    ...
});
```

# Exercise 2: Testing with Jest

You are given a function `rotateRight<T>(arr: T[], k: number): T[]` that returns a new array containing the elements of `arr` rotated to the right by k steps.

- The function must not mutate the input array.
- k must be a non-negative integer.
- Examples:
  - `rotateRight([1,2,3,4], 1)` returns `[4,1,2,3]`
  - `rotateRight([1,2,3,4], 4)` returns `[1,2,3,4]`

Write comprehensive tests for `rotateRight` using Jest.