

UMassAmherst

Manning College of Information
& Computer Sciences

Programming Methodology

Lab 13: Final Review

Wednesday, May 17th, 2023



Weekly Lab Agenda

- Go over reminders/goals
- Review past material
- Work in groups of 2-3 to solve a few exercises
 - Lab leaders will assign new groups this week
- Discussion leaders will walk around and answer questions
- Solutions to exercises will be reviewed as a class
- Attendance taken at the end

Reminders

- Great job this semester everyone, you should be proud of your hard work!
- Max is hosting a final exam review session Sunday December 10 from 7-9pm room TBA, we'll talk about last semester's final exam!
- Office hours will continue to happen as scheduled.
 - Exceptions will be announced on campuswire.
- Exam Logistics:
 - Friday Dec. 15th 3:30pm - 5:50pm in Totman Gym.
 - Make sure to check SPIRE on exam day in case there is a last minute location change!
- Please fill out the [SRTI course survey](#), this really helps make the class better!

Today's Goals

- Practice working with program correctness
- Practice working with asynchronous programming

Exercise: Program Correctness

The following code should partition the given array in-place such that all odd numbers come before all even numbers.

First, write the invariants which satisfy the high-level algorithm.

Then, fill in the code to satisfy the invariants.

```
function partition_even_odd(arr) {  
  if (arr.length === 0) { return; }  
  let low = ???;  
  let high = ???;  
  // low/high form a window, the outside of which is partitioned;  
  // the window shrinks iteratively until everything is partitioned  
  while (???) {  
    if (???) {  
      // swap arr[low] and arr[high]  
      ???  
    }  
    if (???) {  
      // update low  
      ???  
    }  
    if (???) {  
      // update high  
      ???  
    }  
  }  
}
```

Exercise 2: Async

UMassAmherst

Manning College of Information
& Computer Sciences

Write a function `asyncPosMap(arr: T[], f: T => Promise(number): Promise<T[]>`. This function takes a generic array, *arr* and an asynchronous function *f*, and returns a new Promise. That promise should be fulfilled with a new array containing the elements of *arr* that for which *f* resolved to a positive number. The promise should reject if at any point *f* rejects. Ensure that calls to *f* occur asynchronously, by using `Promise.all`.

Fall 2022 Midterm 2 Makeup - 20pts