

# Life after CIS 1100 & Exam Review

# What Comes Next?

- Immediately:
  - CIS 1200
    - More of the same, but you'll use OCaml & Java
  - CIS 1600
    - This is a math course. It's very challenging.
- Later:
  - CIS 2450, Big Data Analytics
  - CIS 2400, Introduction to Computer Systems
  - CIS 1210, Algorithms & Data Structures
  - NETS 1500, Market & Social Systems
  - CIS 19xx Minicourses

## *How to Think about Yourself after 1100*

- Are you a "beginner" or "intermediate" programmer?
- What do you want to do with your programming skills?
- Is it important to learn other programming languages?

# *Getting Involved in Research*

Are you curious about CS research? We have groups in:

- [Programming Languages](#)
- Databases
- Security
- Theory & Algorithms
- Machine Learning & AI
- Natural Language Processing

# *Staying Involved in the Field*

- Find a friend from the class and try to work on a project together
  - Could be about learning something new or just pursuing something you want to explore
  - **Important to stay in touch with the people that you meet in your courses!**
- Join a club or group that does programming
  - PennApps Hackathon, [Crack the Code](#), [Penn Labs](#)
- Apply to be a TA
  - There are a lot of applicants per semester, so don't be discouraged if you don't get in the first time

# Exam Prep

# *How to Prepare*

1. Sit down and take Exam 1 and Exam 2 from this semester
  - Time yourself, 2hr total
  - No notes!
2. Grade yourself and divide your responses into three buckets:
  - i. "I knew this question and got it right or basically right."
  - ii. "I see why I got this wrong."
  - iii. "I have no idea what I was doing here."
3. Congratulate yourself on questions in the first bucket, find practice questions on topics in the second bucket, and review topics in the third bucket.
  - i. "review" == slides, TAs, group study

# Exam Details

- May 10th from 3-5pm
- CHEM 102 (big room!)
- 2 hours, so the length of Exam 1 and Exam 2 put together
- All topics covered are fair game
  - Expect less complexity on topics not covered on homework assignments:
    - 2D arrays, directories, file writing
  - Any topic discussed in lecture is fair game!
- Will try to adjust difficulty down from Exam 2, but you should still prepare for a hard test!



## T.P.S. on Recursion

Given a rectangular 2D array of integers, write a **recursive method** that returns the row and column indices of the first occurrence of a given target integer. If the target integer is not found in the array, return {-1, -1}.

```
public static int[] findElement(int[][] arr, int target) {
    for (int i = 0; i < arr.length; i++) {
        for (int j = 0; j < arr[i].length; j++) {
            if (arr[i][j] == target) {
                return new int[] {i, j};
            }
        }
    }
    return new int[] {-1, -1};
}
```

## *T.P.S. on Recursion*

```
public static int[] findElement(int[][] arr, int target) {  
    // TODO!  
}
```

THINK for 2 minutes:

- Base cases?
- Recursive cases?
- What insights can I gain from the iterative solution?
- Should I think about using helper methods?

## *T.P.S. on Recursion*

```
public static int[] findElement(int[][] arr, int target) {  
    // TODO!  
}
```

PAIR UP for 5 minutes:

- First: share ideas on base cases and recursive cases.
  - What triggers each? What do we do when each is triggered?
- Then: try to write the method together.
  - Have one person write the code and the other person guide them.

## *T.P.S. on Recursion*

```
public static int[] findElement(int[][] arr, int target) {  
    // TODO!  
}
```

SHARE:

Can we collaboratively generate a solution?