

Welcome to Week 1!

CodePath Intermediate Software Engineering

**What's your dream company or
product that you'd want to work
on?**

Feel free to talk in the room chat



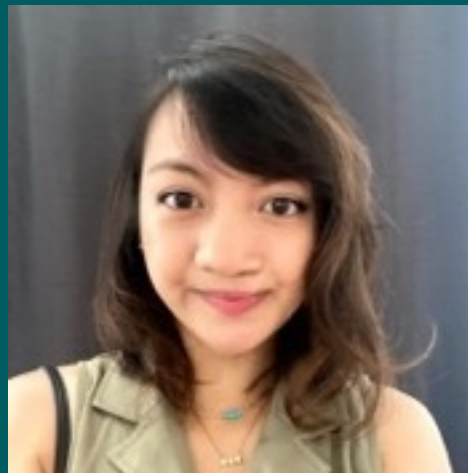
Agenda

- Intros
- Why study programming problems?
- Goals
- Breakout sessions - meet your pod!
- Top 5 coding interview mistakes and how to avoid them
- Free form Q&A!

Instructor Intros!



Caren



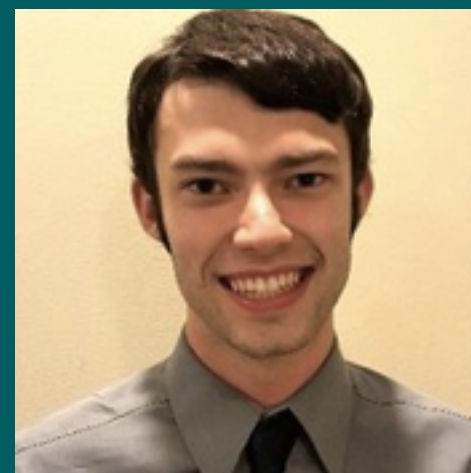
Paulina



Ben



Sushma



Noel

TA Intros!



Allison



Amanda



Best



Geethika



Rashmi



Trang

Coding Interviews are **difficult**

Coding Interviews are **difficult**

Like most things in life,
luck plays a big factor

Coding Interviews are **difficult**

Like most things in life,
luck plays a big factor

Luck is when **preparation** meets
opportunity

Goals for the course

Goals for the course

- Be able to go into phone screens with confidence

Goals for the course

- Be able to go into phone screens with confidence
 - Most questions follow the same pattern!
 - The key is **repeated practice**

Goals for the course

- Be able to go into phone screens with confidence
 - Most questions follow the same pattern!
 - The key is **repeated practice**
- Learn more about the tech industry

Goals for the course

- Be able to go into phone screens with confidence
 - Most questions follow the same pattern!
 - The key is **repeated practice**
- Learn more about the tech industry
- Be prepared to **continue studying and practicing** after the course ends

**How are we going to work
towards the goal?**

How are we going to work towards the goal?

- Lectures to reinforce concepts and walkthrough techniques

How are we going to work towards the goal?

- Lectures to reinforce concepts and walkthrough techniques
- In class exercises for hands-on practice

How are we going to work towards the goal?

- Lectures to reinforce concepts and walkthrough techniques
- In class exercises for hands-on practice
- Extra practice problems and review materials to help you fill potential gaps

How are we going to work towards the goal?

- Lectures to reinforce concepts and walkthrough techniques
- In class exercises for hands-on practice
- Extra practice problems and review materials to help you fill potential gaps
- Weekly HackerRank assessments to keep yourself on track

How are we going to work towards the goal?

- Lectures to reinforce concepts and walkthrough techniques
- In class exercises for hands-on practice
- Extra practice problems and review materials to help you fill potential gaps
- Weekly HackerRank assessments to keep yourself on track
- Mentors, fellow classmates, instructors for help and support



**What is your personal
goal for the next 12 weeks?**

Expectations

Attend weekly sessions
Tuesdays and Saturdays

Complete weekly HackerRank test
*linked in course portal every week

Optional resources: extra reading and practice problems

Course Portal

https://courses.codepath.com/courses/intermediate_software_eng

The screenshot displays the 'Intermediate Software Engineering' course portal. At the top, the course title is prominently displayed next to a help link: 'Need help? Post on our discussion system or email us at support@codepath.org'. Below the title, a navigation bar includes links for 'Overview', 'Session #1', 'Session #2', 'Assignment', 'Resources', and 'Organizer'. A left-hand sidebar lists weeks from 'Week 1' to 'Week 12', with 'Week 1' currently selected and highlighted in blue. The main content area for 'Week 1 - Warmup / UMPIRE' features a sub-header, a brief introduction, a 'Goals' section with four bullet points, and an 'After the session' section. The 'After the session' section includes a deadline: 'Due before Monday, June 10th at 12:00am PDT' and a paragraph describing a Hackerrank assessment with optional practice problems, concluding with the instruction: 'Plan to spend 90 minutes on this assessment.'

Intermediate Software Engineering Need help? Post on our discussion system or email us at support@codepath.org

Overview Session #1 Session #2 Assignment Resources Organizer

Overview

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

Week 10

Week 11

Week 12

Week 1 - Warmup / UMPIRE

This week, we will do some orientation, learn about the UMPIRE method, and start practicing simple interview questions!

Goals


- Get familiar with the format and goals of the course
- Practice a common behavioral interview question
- Learn about the UMPIRE method
- Practice using the UMPIRE method to solving coding problems.

After the session

Due before Monday, June 10th at 12:00am PDT

This assignment for this week is a Hackerrank assessment that covers topics we went over in class for the week. Check out the [assignment tab](#) for the Hackerrank link. We also selected a few optional practice problems that helps supplement what we covered in class. While the problems are optional, we highly encourage you to try them out before taking the Hackerrank assessment as practice. The HackerRank assessment will be a mix of multiple choice and coding challenges. **Plan to spend 90 minutes on this assessment.**

Course Portal



Intermediate Software Engineering

Need help? Post on our [discussion system](#) or email us at support@codepath.org

[Overview](#)[Session #1](#)[Session #2](#)[Assignment](#)[Resources](#)[Organizer](#)

[Week 1](#)[Week 2](#)[Week 3](#)[Week 4](#)[Week 5](#)[Week 6](#)[Week 7](#)[Week 8](#)[Week 9](#)[Week 10](#)[Week 11](#)[Week 12](#)

Week 1 - Warmup / UMPIRE

This week, we will do some orientation, learn about the UMPIRE method, and start practicing simple interview questions!

Goals

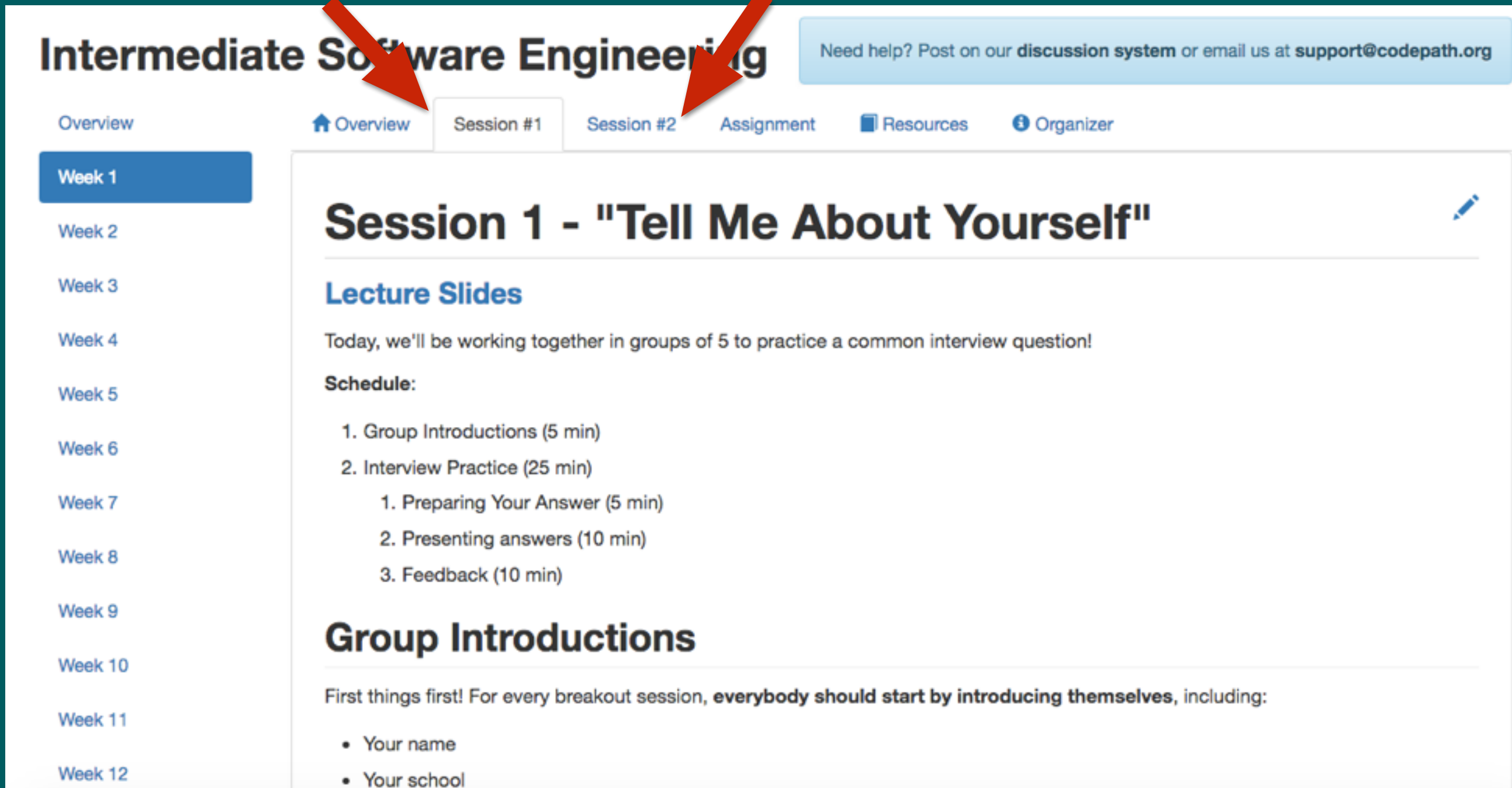
- Get familiar with the format and goals of the course
- Practice a common behavioral interview question
- Learn about the UMPIRE method
- Practice using the UMPIRE method to solving coding problems.

After the session

Due before Monday, June 10th at 12:00am PDT

This assignment for this week is a Hackerrank assessment that covers topics we went over in class for the week. Check out the [assignment tab](#) for the Hackerrank link. We also selected a few optional practice problems that helps supplement what we covered in class. While the problems are optional, we highly encourage you to try them out before taking the Hackerrank assessment as practice. The HackerRank assessment will be a mix of multiple choice and coding challenges. **Plan to spend 90 minutes on this assessment.**

Course Portal



Intermediate Software Engineering

Need help? Post on our [discussion system](#) or email us at support@codepath.org

Overview Session #1 Session #2 Assignment Resources Organizer

Week 1
Week 2
Week 3
Week 4
Week 5
Week 6
Week 7
Week 8
Week 9
Week 10
Week 11
Week 12

Session 1 - "Tell Me About Yourself"

Lecture Slides

Today, we'll be working together in groups of 5 to practice a common interview question!

Schedule:

1. Group Introductions (5 min)
2. Interview Practice (25 min)
 1. Preparing Your Answer (5 min)
 2. Presenting answers (10 min)
 3. Feedback (10 min)

Group Introductions

First things first! For every breakout session, **everybody should start by introducing themselves**, including:

- Your name
- Your school

Course Portal

Intermediate Software Engineering

Need help? Post on our [discussion system](#) or email us at support@codepath.org

Overview

Overview

Session #1

Session #2

Assignment

Resources

Organizer

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

Week 10

Week 11

Week 12

Assignment 1

  Submit

Due before Monday, June 10th at 12:00am PDT

HackerRank

The assignment for the week is to complete a HackerRank assessment. The assessment will contain a mix of multiple choice questions and coding problems and should take around 90 minutes.

Post-session review / practice

- [UMPIRE Guide](#)
- Check out the Resources tab for session slides, exercise solutions, and additional links

Extra practice problems, not required

- <https://projecteuler.net/problem=1>
- <https://projecteuler.net/problem=2>

Note: If you sign up for an account on Project Euler, you can verify your answers.

Course Portal

Intermediate Software Engineering

Need help? Post on our [discussion system](#) or email us at support@codepath.org

Overview

Overview

Session #1

Session #2

Assignment

Resources

Organizer

Week 1

Week 2

Week 3

Week 4

Week 5

Week 6

Week 7

Week 8

Week 9

Week 10

Week 11

Week 12

Resources

Lectures

- Session 1 - Recording, Slides
- Session 2 - Recording, Slides, Exercise Solution

Guides

- [UMPIRE Guide](#)

Additional Links

- [Project Euler \(practice problems\)](#)
- [Tips and techniques for the coding interview](#)

Pods / Teams

Pods / Teams

- 5-6 students per team
- This is the group you'll be working with for the next 12 weeks
- Help support each other throughout the course!

Pods / Teams

- Find your pod number, prepend it to your name right now! (Caren to send link in Zoom chat)
- Students: 5 - Caren Chang
- Mentors: 4, 5 - Caren Chang

Mentors

- Engineers working in the tech industry - some are past students of this course!
- Will be there for most of your Saturday sessions to help you through in-class exercises

In class exercises

In class exercises

- Break out into small groups of 5-6
- Work on problems together for the topic of the week
- Great practice for phone interviews!

Our first in class exercise!

Our first in class exercise!

- Most common interview question: “Tell me a little about yourself”

Our first in class exercise!

In a moment, we're all going to into small groups with our pods

Our first in class exercise!

In a moment, we're all going to into small groups with our pods

Start by having everybody introduce themselves in alphabetical order

Our first in class exercise!

In a moment, we're all going to into small groups with our pods

Start by having everybody introduce themselves in alphabetical order

Take 5 minutes to prep for your answer (guidance in course portal).
Your 'spiel' should be ~3 minutes

Our first in class exercise!

In a moment, we're all going to into small groups with our pods

Start by having everybody introduce themselves in alphabetical order

Take 5 minutes to prep for your answer (guidance in course portal).
Your 'spiel' should be ~3 minutes

Take turns answering the question, jot down notes if you're not the one speaking

Our first in class exercise!

In a moment, we're all going to into small groups with our pods

Start by having everybody introduce themselves in alphabetical order

Take 5 minutes to prep for your answer (guidance in course portal).
Your 'spiel' should be ~3 minutes

Take turns answering the question, jot down notes if you're not the one speaking

Once everyone is done, get some feedback!



Common Mistakes

Common Mistakes

- Not mentioning why you're interested in the role / company

Common Mistakes

- Not mentioning why you're interested in the role / company
- Only talking about things that's already on your resume

Common Mistakes

- Not mentioning why you're interested in the role / company
- Only talking about things that's already on your resume
- Not showing enthusiasm

Common Mistakes

- Not mentioning why you're interested in the role / company
- Only talking about things that's already on your resume
- Not showing enthusiasm
- Not knowing about the company / role you're interviewing for

Great things to talk about

- Interesting focuses / projects from past companies
- Passion projects
- Why you're interested in the company
- Specialities you're interested in (Mobile, Machine Learning, ...)

Ready for our first breakout rooms?

- For instructions, go into the course portal, Week 1 - Session 1
- Let's meet back at 4:25pm (Pacific), 7:25pm (Eastern)

How'd it go?

**The key to acing
interviews**

The myth: I have to solve 300+ questions on Leetcode and memorize solutions in order to be successful in interviews

The myth: I have to solve 300+ questions on Leetcode and memorize solutions in order to be successful in interviews

The reality: Most problems can be solved following the same patterns / tricks. You should only have to study about 30-40 problems really well in order to be successful in 90% of coding interviews.

The myth: I have to get every problem correct in the most optimal way in the fastest time frame to pass

The myth: I have to get every problem correct in the most optimal way in the fastest time frame to pass

Top Common interview mistakes:

- not communicating with interviewer

The myth: I have to get every problem correct in the most optimal way in the fastest time frame to pass

Top Common interview mistakes:

- not communicating with interviewer
- jumping to conclusions / solving the wrong problem

The myth: I have to get every problem correct in the most optimal way in the fastest time frame to pass

Top Common interview mistakes:

- not communicating with interviewer
- jumping to conclusions / solving the wrong problem
- doesn't catch or can't fix bugs

The myth: I have to get every problem correct in the most optimal way in the fastest time frame to pass

Top Common interview mistakes:

- not communicating with interviewer
- jumping to conclusions / solving the wrong problem
- doesn't catch or can't fix bugs
- not discussing space / run time tradeoffs

The myth: I have to get every problem correct in the most optimal way in the fastest time frame to pass

Top Common interview mistakes:

- not communicating with interviewer
- jumping to conclusions / solving the wrong problem
- doesn't catch or can't fix bugs
- not discussing space / run time tradeoffs
- speed - interviewer wasn't able to get a good signal

U

M

P

I

R

E

Understand

Match

Plan

Implement

Review

Evaluate

Understand

Understand what the interviewer is asking with clarifying questions and test cases

Understand

Understand what the interviewer is asking with clarifying questions and test cases

State any assumptions you make

- is the input always sorted?
- is the input guaranteed to satisfy x & y conditions?

Understand

Understand what the interviewer is asking with clarifying questions and test cases

State any assumptions you make

- is the input always sorted?
- is the input guaranteed to satisfy x & y conditions?

Given... x input, do we expect y output?

Match

Does this problem match any common patterns we've seen?

Which data structures / techniques can we use to simplify this problem?

- can we use hash tables for easy lookup later?
- would using stacks / queues be helpful?
- should we use Depth First Search or Breadth First Search?

Plan

It's easier to modify your solution before you write all the code

Plan

It's easier to modify your solution before you write all the code

Catch potential bugs before starting to write code

Plan

It's easier to modify your solution before you write all the code

Catch potential bugs before starting to write code

Use diagrams and pseudocode to visualize how the problem will be solved

Plan

It's easier to modify your solution before you write all the code

Catch potential bugs before starting to write code

Use diagrams and pseudocode to visualize how the problem will be solved

Run through your approach with test cases to check that it works

Implement

Code!

Review

Trace through your code with an input to check for the expected output

Catch possible edge cases and off-by-one errors

Evaluate

Analyze the run time and space complexity of your solution

Discuss tradeoffs that were made, or assumptions that were taken

Questions?

Next Session

Walk through a problem with UMPIRE approach

Group exercise to try UMPIRE yourself!

Survey