Learn About the Company

Website (university / career page)

GlassDoor

Annual Report

Catherine

Learn About the Position

More than just “Programming”

Ask about job desc.

If none, ask about languages, frameworks, methodologies

Learn About the Interview

Ask about format (whiteboard, screen share, shared terminal, etc)

Be prepared to ask meaningful questions!

Questions mean you’re engaged

Make sure you know the things you claim

Worked with something once? Twice? Nope.

Examples!

Try explaining to a non-CS person to communicate basics

Coursework

Everyone knows C / C++

What matters is how you can apply it

Research

personal projects

internships

separate yourself from 80% of students

theory or not, use pratical examples

talk about classes you ENJOYED and LEARNED the most

talk about your most challenging problem

review old coursework to stay fresh in mind

Projects

scope, scope scope

goals, customers, accomplishments, failures, tools

My questions

Not trick questions, puzzles are ok

not language-specific - we care about the higher level concept

problem solving > language skills

connected to a real life problem that the engineer had to solve

logic correctness > syntax

Why coding questions?

verify coding skill

interact with people

validation of coding skills (tests, how do we know we are correct?)

Non-coding questions

Open-ended, your time to provide info to me

anything on your resume is fair game

What would you like to work in? no answer is bad

Pitfalls

not thinking through

not asking clarification questions

not being specific in the answer

not answering at all (stalling)

assuming short-answers are best (interviewer drives the time)

coding: not talking out loud

being silient when stuck (describe your situation)

not making sure it works correctly

reverse a string

palidrome

describe your last big project - how work? what tech? design?

How would you move mnt. fuji? - Will Poundstone

cracking the coding interview - Gayle Lakman