Welcome to CS 301

Foundations of Computer Science

Slides derived from my lecture notes with the help of Claude Opus 4.1

Course Overview

Two Interconnected Worlds

🧮 Computational Theory

What are the fundamental **capabilities** and **limitations** of computers?

⚙️ Computational Practice

How to solve problems using **industry-standard** approaches?

The Balancing Act

Like Philippe Petit on a Tightrope...

We'll balance theory and practice through:

1. **Drawing problems from theory**
   - Example: Build an arithmetic expression interpreter
2. **Focusing on practical applications**
   - Example: Implement regular expression matching

World 1: Computational Theory

Three Fundamental Areas

1. **Automata and Formal Languages** 🤖
   - What IS a computer? What IS a computational problem?
2. **Computability** 🎯
   - What CAN we solve? What's IMPOSSIBLE to solve?
3. **Complexity** ⏱️
   - What's TRACTABLE? What's INTRACTABLE?

Our Theory Approach

Focus on Insights, Not Proofs

✅ What we'll emphasize:

- **Conclusions** and their real-world relevance
- **Practical implications** of theoretical limits
- **Intuitive understanding** of concepts

📚 Helpful background (but not required!):

- Data structures (stacks, queues, graphs)
- Basic mathematics (sets, functions, logic)

> *Missing something? No worries! We'll cover it together.*

World 2: Computational Practice

Industry-Standard Approaches

🤝 Collaboration

- Version control (Git/GitHub)
- Parallelized development
- Code reviews

🚀 Performance

- Algorithm selection
- Memory management
- Profiling and optimization

💻 Modern Tools

- Terminal/command line mastery
- Generative AI for development

How We'll Practice

Learning by Doing

- **Individual assignments** - Build your skills
- **Group projects** - Learn to collaborate
- **In-class exercises** - Apply concepts immediately
- **"Tales from the Trenches"** - Real industry stories

Sneak Peek: Profound Conclusions

Three Big Ideas We'll Explore

1. Universality 🌍

All computers are created equal!

Your laptop ≈ Your phone ≈ Supercomputer ≈ **Turing Machine**

They all have the same fundamental computational power

(just different speeds and memory)

2. Computability 🚫

Some problems are impossible to solve

No computer, no matter how powerful, will EVER solve certain problems

Example: The Halting Problem

- Can we write a program that determines if any program will halt or run forever?
- **Answer:** Provably impossible!

3. Complexity 🤯

Some solvable problems might be practically impossible

There exist problems where:

- We CAN solve them (in theory)
- But it might take longer than the age of the universe
- We don't know if there's a faster way!

Example: Traveling Salesman Problem for large number of cities

📝 Reflection Activity

1. **One thing** that excited you about this course
2. **One concern** you have
3. **One question** you want answered

🚀 Let's Begin!