Welcome to CIS 210: Computer Science I. This course covers basic concepts and practices of computer science. Topics include algorithmic problem solving, levels of abstraction, object-oriented design and programming, software organization, analysis of algorithm and data structures.

Prerequisites: MATH 112. Prior programming experience strongly encouraged.
Short Biography

• Ph.D. in Mathematics with an emphasis in Computational Science and Engineering from UC Santa Barbara, 2010

• Postdoctoral fellowship at Stanford University (Geophysics)

• Computational seismologist (postdoctoral researcher) at San Diego State University

• Assistant Professor in Mathematics and Statistics, Portland State University, 2014-2018

• Assistant Professor in Computer and Information Science (Joint with ERTH), University of Oregon, 2018-present
My research integrates applied mathematics, computer science and geophysics to solve outstanding problems in earthquake science.
My research integrates applied mathematics, computer science and geophysics to solve outstanding problems in earthquake science.

*This is the first time I have taught CIS 210. Please ask questions at any point during lecture. There may be times I might have to answer with “I don’t know, but I assure you I will find out and report back to you.”*
• What is Computer Science?
• Computational Problem Solving
• What can you expect from CIS 210?
• Prior programming experience/quiz

Course Website: https://classes.cs.uoregon.edu/20W/cis210/
What is Computer Science?

- the study of algorithms (MR p. 2)
- the study of problems, the [computational] problem-solving process, and the solutions [algorithms, programs] to those problems. (MR p. 330)
Computational problem solving is an *algorithms-based* approach to problem solving that is *inspired and constrained* by the *possibilities and limitations* of computers and computing.
An **algorithm** is a sequence of well-defined operations. FOR EXAMPLE: Fizzbuzz

0. form into groups of 3-5 students
1. the first person says the number 1
2. go around the group, with each person saying the next number in turn
3. though if the number is divisible by 3, say “fizz”, and if the number is divisible by 5, say ”buzz”, and if the number is divisible by 3 and 5, say “fizzbuzz”
4. if an error is made, start again
5. stop when you reach 100
ALGORITHMS (structured solutions to problems)

- have been around for a long time
  – use an existing one
  – adapt (revise, refactor) an existing one
  – develop a new one

-can be carried out (implemented) by a person or a computer
ALGORITHMS (structured solutions to problems)

- have been around for a long time
  – use an existing one
  – adapt (revise, refactor) an existing one
  – develop a new one

-can be carried out (implemented) by a person or a computer

Computational process: an algorithm that can be implemented on a computer.
COMPUTATIONAL PROBLEM SOLVING

PROBLEM/TASK ➔

ALGORITHM ➔

COMPUTATIONAL PROCESS ➔

COMPUTER PROGRAM
Computational Problem Solving

TASK/PROBLEM $\leftrightarrow$ Computational Thinking $\leftrightarrow$ ALGORITHM/COMPUTATIONAL PROCESS

ALGORITHM $\leftrightarrow$ Design/Coding $\leftrightarrow$ COMPUTER PROGRAM

COMPUTER PROGRAM $\leftrightarrow$ Testing/Debugging $\leftrightarrow$ HIGH QUALITY COMPUTER PROGRAM

HIGH QUALITY COMPUTER PROGRAM $\rightarrow$ Execute $\rightarrow$ AUTOMATIC, FAST, RELIABLE, REUSABLE COMPUTER-GENERATED SOLUTION
Enlisting a computer as a problem-solving partner requires addressing the limitations of computers
Enlisting a computer as a problem-solving partner requires addressing the limitations of computers.
Enlisting a computer as a problem-solving partner requires addressing the limitations of computers 0, 1

We need tools, skills, approaches for communicating with computers for computational problem solving.

science, math, design, engineering
CIS 210 Computer Science I

What is Computer Science?

Support for/computational problem solving

Theory
• automata theory
• algorithms & data structures
• complexity science, math, engineering, design
• programming languages

Systems
• computer organization
• operating systems
• networks/high performance computing/security

Software Development/Engineering
• programming best practices
• programming large, complex systems

Applied Computer Science
• data analytics
• computing + X (e.g., biology, linguistics, law, economics, etc.)
✓ What is Computer Science?

✓ Computational Problem Solving

• What can you expect from CIS 210?

• Prior programming experience/quiz

Course Website: https://classes.cs.uoregon.edu/20W/cis210/
AFTER COMPLETING CIS 210 COMPUTER SCIENCE I, YOU WILL BE ABLE TO …

- understand, develop, and implement algorithms for **computational problem solving**;

- use **structured design and testing methods** to develop and implement computational solutions to problems (programs);

- read, write, revise, document, test, and debug algorithms and code;

- demonstrate robust mental models of data representation and code execution;

- demonstrate good understanding of a high level programming language;

- introduce and/or implement a sampling of classic computer science problem domains and algorithms.
What can you expect from CIS 210

CIS 210 Focus: Computational Problem Solving for a depth and breadth intro to computer science

- Expand/improve software development skills
- Demonstrate robust mental models of data representation and code execution
- Introduction to Computer Science topics
CIS 210 Computer Science I

What can you expect from CIS 210?

✓ Weekly projects and exercises to support learning of computational problem solving and other computer science topics

Supported by
✓ Class – large group meetings, concepts, discussion, clickers
✓ Lab (KLA B26) – small group exercises, computers
✓ Class notes, project solutions, text readings and exercises – review, practice, explore
✓ Lab help hours – daily drop-in help
✓ Code review – 2+ per term
✓ Code demonstration – 1 per term
✓ Assessments – midterm and final exams
✓ Class Encore – weekly guided study groups
CLASS ENCORE

Small study groups for challenging classes allow students to gather once a week outside of class to practice course concepts and strategies. Peer leaders attend the classes and design collaborative activities for each group meeting.

CIS 210

**THURSDAYS**
11-11:50am with Ellie
12-12:50pm with Ellie

**FRIDAYS**
11-11:50am with Sam
12-12:50pm with Sam

Sign up at classencore.uoregon.edu
CIS 210 Computer Science I

Hours per week:

- Textbook reading/practice: 1-2
- class/lab: 4
- weekly projects – understand/design/implement/test&debug: 6+ [lab help hours]
- review text/class & lab notes/project solutions: 1-2
What is Computer Science?

Computational Problem Solving

What can you expect from CIS 210?

- Prior programming experience/quiz
REVIEW ALL OF THE MATERIAL AT CLASS WEBSITE!

Required text / on reserve in library.

Schedule page – Weekly topic and readings, Class notes, Lab exercises, Projects, Project Solutions, Practice Exams, Exams, Solutions

Syllabus – Class policies, how final grade is calculated, much more

Other Information – Lab help hours schedule, contact information, helpful and interesting links

Course Website: https://classes.cs.uoregon.edu/20W/cis210/
CIS 210 Computer Science I

I will be gone at a workshop this Thursday, January 9th. GE Sam Schwartz will be lecturing that day.

*Because I am traveling this week my office hours (this week only) will be postponed to Friday. I will post the new office hours on Canvas and on the course website later today.

QUIZ

Course Website: https://classes.cs.uoregon.edu/20W/cis210/