CIS 122
Introduction to Programming and Problem Solving
Spring 2017

Course Description

CIS 122 is an introduction to computational problem solving, algorithm design, and programming in Python. The class introduces topics in computational thinking, including techniques for program design, implementation, and testing. Regular class meetings are supplemented with weekly computer labs.

This course is intended for students with no prior programming experience in any language. CIS 122 is appropriate for students from any major who are interested in a practical, one course introduction to computing. It is also appropriate for students interested in gaining programming experience prior to beginning the computer science introductory sequence for majors.

CIS 122 is in the science group, and can count toward the Bachelor of Science mathematics/computing requirement.

Programming and Problem Solving

Computational Problem Solving
Computational Problem Solving

Get the computer to do the work for us

✓ automatic
✓ fast
✓ reliable
✓ reusable

solutions to a wide variety of tasks

How can we do this?
Get the computer to do the work for us
How can we do this?
There’s an app for that.

CIS 122:
Using programs → Creating programs

Enlisting a computer as a problem-solving partner requires addressing the limitations of computers
CIS 122: Using programs → Creating programs

Enlisting a computer as a problem-solving partner requires addressing the limitations of computers

0, 1
Enlisting a computer as a problem-solving partner requires addressing the limitations of computers

0, 1

- Tools (e.g., language) for communicating
- Computational approach to problem-solving

Computational problem solving is an approach to problem solving that is inspired and constrained by the possibilities and limitations of computers and computing
Programming as Computational problem solving:

Start with a task (for example, a calculation to perform, an idea to implement, a domain to explore, a problem to solve, etc.) and apply a computational process.
Programming as Computational problem solving:

Start with a task (for example, a calculation to perform, an idea to implement, a domain to explore, a problem to solve, etc.) and

apply a computational process

to obtain a desired outcome.
A computational process is a sequence of well-defined operations that leads from an initial starting point to a desired final outcome.

FOR EXAMPLE: How many students are in class today?
FOR EXAMPLE:
How many students are in class today?

1. everyone stand up
2. assign yourself the # 1
3. while the # of standing students is > 1 –
   a. partner with the person closest to you
   b. add your numbers
   c. one person is assigned the sum
   d. the other person sits down
4. report the total.

A sequence of well-defined operations is called an algorithm.
FOR EXAMPLE:
How many students are in class today?

COMPUTATIONAL PROBLEM SOLVING:
• desired ending point
• initial starting point
• description of how to move systematically from the input to the output/algorithm

ALGORITHMS

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 machine
Other examples of algorithms?

TASK $\leftrightarrow$ Computational Thinking $\leftrightarrow$

COMPUTATIONAL PROCESS/ALGORITHM $\rightarrow$

DESIRED OUTCOME
Computational thinking is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent. [CunySnyderWing10]

A computer program implements an algorithm on a computer.
Coding is a process that leads from an algorithm to an executable program.

Designing, writing, documenting, testing, debugging

Programming language
CIS 122 Introduction to Programming and Problem Solving  Spring 2017

TASK $\leftrightarrow$ Computational Thinking $\leftrightarrow$
ALGORITHM $\leftrightarrow$ Coding $\leftrightarrow$
COMPUTER PROGRAM $\rightarrow$
DESIRED OUTCOME

*Programming = Computational Thinking + Coding*

CIS 122 Introduction to Programming and Problem Solving  Spring 2017

Why (should you learn to) program?

(0) Gain fluency in computing which is ubiquitous in our lives today.

“Program or be programmed.”
-- Rushkoff
(1) Programming as a liberal art:

Programming is a fun and useful intellectual exercise which develops or enhances valuable transferable computational thinking skills, including logical, creative, design skills.

-- Campbell, et. al.

(2) Gain a tremendously powerful problem tool.

“Advances in computing have expanded our capacity to solve problems at a scale never before imagined, using strategies that have not been available to us before.”
Why (should you learn to) program?

(3) Programming is a fundamental part of computer science for minors/majors/professionals, and provides exposure to many topics in the field of computer science.

(4) CIS 122 counts toward UO B.S. math/computing requirement, UO science group requirement, and as a CIS 210 programming prerequisite.

Which CIS course is right for you?

105 Explorations in Computing
110 Fluency with Information Technology
111 Introduction to Web Programming
115 Multimedia Web Programming
122 Introduction to Programming & Problem Solving
210, 211, 212 Computer Science I, II, III
CIS 122 Introduction to Programming and Problem Solving Spring 2017

CIS 122
https://classes.cs.uoregon.edu/17S/cis122/index.html

(Canvas is a portal to the class website.)

Large group
Labs Projects (weekly)
Lab help hours Quizzes/Exams
Textbook/Exercises
Python/IDLE
+ Links at class website

CIS 122 Introduction to Programming and Problem Solving Spring 2017

TASK ←→ Computational Thinking ←→
ALGORITHM ←→ Coding ←→
COMPUTER PROGRAM → Execute → SOLUTION

Programming = Computational Thinking + Coding
-------------

A computer program implements an algorithm on a computer.

A computer program is (therefore) a set of instructions written in a language the computer can understand.
A program is a set of instructions written in a language the computer can understand.

**WHICH language is that?**

Natural language?
Flip switches?
Programming language