CSE131 — Computer Science I

Prof. Ken Goldman
Jolley Hall — Room 512
Office Hours: Mon. 11:15-1 + Th. 11:15-12:15
kigel@wustl.edu
935-7542

Computer Science I

Not a Natural Science
Natural Science asks: Why? What?
Computer Science asks: How?

This class is not really about computers

Computer Science is about correct and efficient automation of process
So the heart of computer science is devising algorithms — ways to carry out processes.

Example: The Soccer Cancelation Problem

Given: Coach C and n players P1, P2, ..., Pn
Everyone has a team roster with names & phone #s.

Want: A good algorithm that lets the coach notify all players that the game is canceled.

Algorithm 1) Brute Force — Coach calls all the players

\[ C \rightarrow P_1 \rightarrow P_2 \rightarrow \ldots \rightarrow P_n \]

Advantages: • Simple
• Coach knows when the job is done

Disadvantages: • Coach does all the work
• Slow! Last player is notified after n phone calls.
Algorithm 2) Call chain

- Advantages: • Spreads the work (each person makes at most one phone call)
  • Still fairly simple

- Disadvantages: • The coach doesn't know when the process is complete.
  • Still slow
  • Worse, someone might not be home, breaking the call chain.

Variation 2b) Like alg. 2, but Pn calls Coach. (Cycle)

Adv: Now the coach will know when the process is complete.

But... what if the chain is broken.
Variation 2c) Like 2b, but if someone isn’t home, call
the next person on the list.

Adv: Fixes the broken chain problem.
Disadv: • If many in a row aren’t home, someone does a
lot of work.

• Still slow!

Let’s think of a different approach...
Algorithm 3) Call Tree
Coach calls 2 people, and so on...

Advantages:
• Spread the work (each person makes at most 2 calls)
• Fast! Even P14 is notified after only 6 calls. (P30 would be notified after only 8.)

Disadvantages:
• A bit more complicated
• If someone isn’t home, a whole subtree might not be notified in time.
• Coach doesn’t know when the process is complete.
Variation 3b) Your parent is the person who called you. Call your parent when everyone below you has been notified.

For example, P2 calls C after both P5 and P6 have called P2.

Advantages: * Still spreads the work and is fast
  * Coach knows when the process is complete

Disadvantages: * Still could miss a whole subtree, but could make further modifications to deal with the case when no one is home...
The soccer cancelation problem is an example of an algorithmic problem. Our discussion was an example of algorithmic thinking: consider possible algorithms + variations, analyze their properties, and ultimately choose the "best" one.

To automate algorithm execution using a computer, we need a way to express the algorithm precisely (unambiguously), concisely, and as naturally as possible. This is the role of a programming language.
Java is the programming language we'll use in CSE131. Java is **object-oriented**, which means we decompose systems into collections of objects, where each type of object belongs to a **class**

We extend the language by defining new classes (new types of objects).

In the above example, we might define objects for the coach, each player, the roster, etc.

In an internet banking application, we might define objects of type `BankAccount`, `CreditCard`, etc.
Each class (type of object) has 2 main parts:

1. instance variables — these hold the data inside of each object
2. methods — these describe what an object can do (including the algorithm for how it will do it)

As a first glimpse of Java, here's a simple class for a BankAccount type.
public class BankAccount {
    // instance variables:
    int balance = 0;

    // methods:
    int deposit(int amount) {
        balance = balance + amount;
        return balance;
    }
}