Introduction

CSE 131 - Computer Science I
Prof. Kenneth J. Goldman (Ken)
512 Jolley Hall
kjh@wustl.edu, (314) 935-7542
www.cse.wustl.edu/~kjh/cse131

Office hours:
Mondays & Thursdays 11:15-12:30
and by appointment
Computer Science I

What is CS?

Techniques for creating correct & efficient software

Examples:

- Abstraction – hiding detail
- Decomposition / Composition
- Inheritance
idea \[\xrightarrow{\text{requirements}}\text{specification}\] \[\xrightarrow{\text{design}}\text{(components & relationships)}\] internet banking

security: passwords?
store user info
convenient

User can:
- check balance
- make a deposit...
user run
compile
Java source code
Each object in the design has its own specification:
- an interface — things we can ask it to do
- legal behaviors — how should it respond
Spec. of Account:

- `getBalance` \( \Rightarrow X \)
- `deposit Z` \( \Rightarrow Y \)
- `getBalance` \( \Rightarrow Y \)
- `Expect: Y = X + Z` 

Spec. does NOT say how to build it:

- `deposit Z` 
- `getBalance` 

**Implementation**

- What info is inside
- How do we perform the methods
Why not express the implementation in English?

Problem: Natural language is ambiguous.

“Add 1 can of beans.”

Programming Language ⇒ UNAMBIGUOUS

Syntax — what you say —
how you say things

Semantics — what it means
Java is object-oriented:
program runs as interacting objects
A type of object is called a class
an object of a class is an instance.
public class BankAccount {
    name of the class
    // instance variables:
    private int balance;
    Type name of variable (identifier)
    // methods:
    public int getBalance() {
        begins method body
        returnType
        return balance;
        return statement
    }
}
public void deposit(int amount) {
    balance = balance + amount;
}