A look at typical CS1 textbooks

- **Chapter 1: Primitives & Arithmetic**
  - Character Strings
  - Printing to the console using println
  - Printing to the console using print
  - String concatenation
  - String escape sequences (Chart)
  - Variables
  - Assignment using =
  - Constants
  - Integer types
    - byte
    - short
    - int
    - long
  - Floating point types
    - float
    - double
  - Characters
    - char
  - Booleans
    - boolean
  - Arithmetic Operators
    - + * / %
  - Operator Precedence
  - Increment/Decrement Operators
  - Assignment Operators
  - Casting and coercion

- **Chapter 2: Introduction to Objects**
  - String class
  - Math class
  - Enumerated Types
  - Instance Variables
  - Constructors
  - Methods
  - Parameters
  - Static

- **Chapter 3: Conditionals**
  - Equality Operators
  - Relational Operators
  - Logical Operators
  - if
  - if-else
  - nested if
  - Ternary operator ?:
  - switch-case

- **Chapter 4: Loops**
  - while
  - do-while
  - for
  - Enhanced for loop
  - break
  - continue

- **Chapter 5: Arrays**
  - Single dimension
    - Creation
    - Indexing
  - Two dimensions
    - Creating
    - Indexing
  - Passing as parameters
  - Storing objects in arrays

- **Chapter 6: Inheritance**
  - Polymorphism

- **Chapter 7: File I/O**

- **Chapter 8: Exception Handling**

- **Chapter 9: Recursion**

- **Chapter 10: GUIs**

- **Chapter 11: Networking**

---

Is this really what we want to teach?

Maybe we should teach this:

- How to solve CS problems
- Principles of software development
- Design & Design Patterns
- Modeling
- Abstraction
- OO concepts:
  - Polymorphism, encapsulation
- Sequencing/selection/repetition

What do **you** think we should teach?

---

Our proposal:

**Design-driven approach**

Start with OO fundamentals:
- Objects (properties & capabilities)
- Object communication (messages)

Focus on Modeling

Teach design principles
- Class relationships
  - dependency, association, composition, realization (implementation), generalization (inheritance)
- Polymorphism
- Encapsulation / Information hiding
- Design patterns

Introduce syntax as needed to do the above