

Why Study Computer Science?



Certificate

Actionable Skills

Work-Ready Graduate

Immediate Payoff

Knowledge

Deep Understanding

Adaptable Graduate

Long-Term Payoff

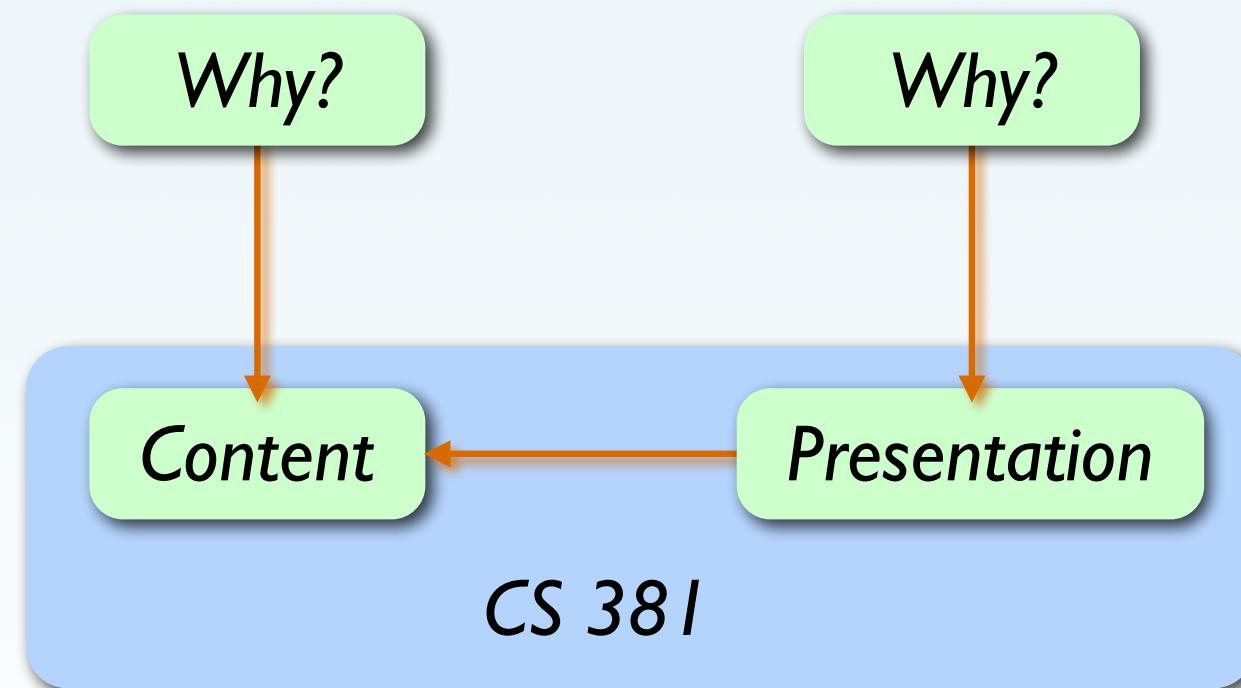


CS 381

How To Be Successful in CS 381

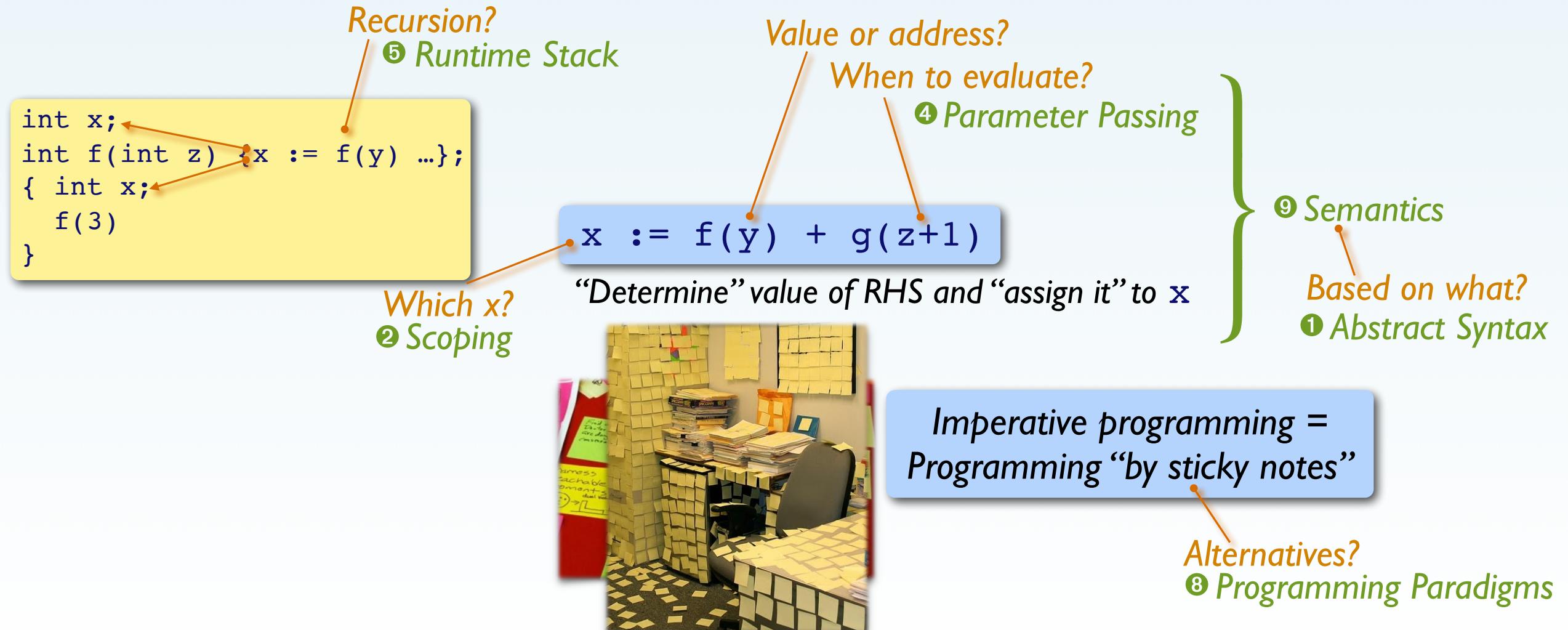
- Take the quizzes seriously!
- Take the homework seriously!
- Start programming in Haskell NOW!
- Use Piazza to ask (*and answer*) questions!

The What, How, And Why



What is CS 381 About?

*What **exactly** is a programming language?*



What About Haskell?



Certificate
Actionable Skills
Work-Ready Graduate
Immediate Payoff

Use of Haskell in Industry



Google facebook

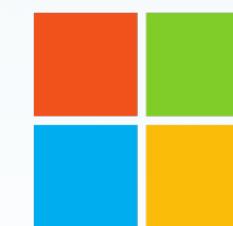
The New York Times

Qualcomm

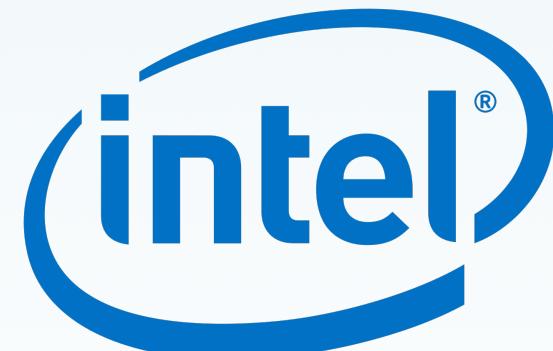
Standard
Chartered



nVIDIA®



Microsoft



The Role of Haskell in CS 381

- **Example** of a non-imperative programming paradigm (just like Prolog)
- **Tool** for describing language concepts (syntax, semantics, scope, typing)

	Metalanguage		
	English	Math	Haskell
Precise	✗	✓	✓
Checkable	✗	✗	✓
Executable	✗	✗	✓



Haskell?

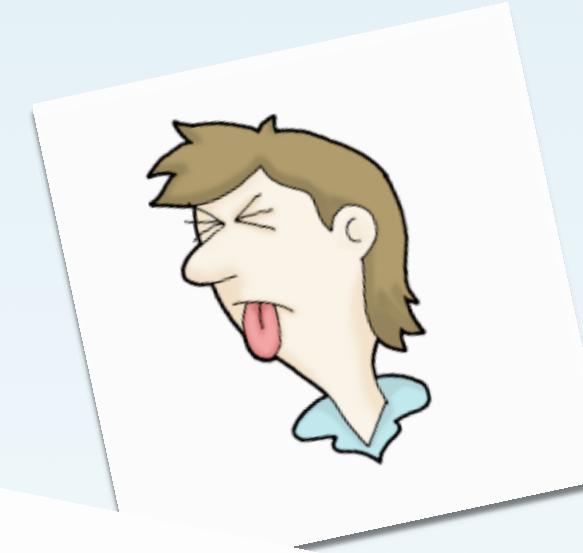


Haskell

Research in PL ?

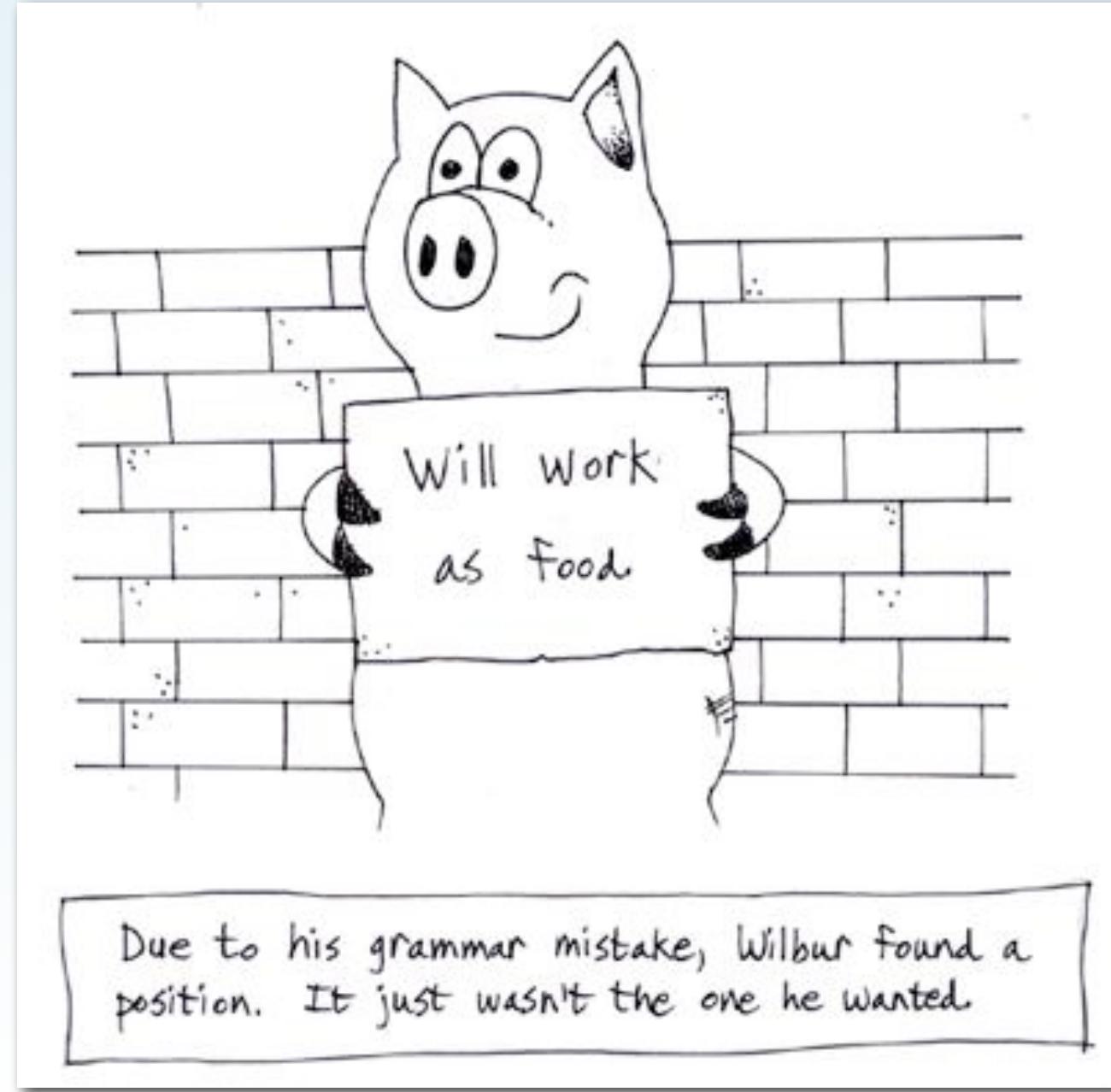
Research Experience as Undergrad ?

Grad School ?



I Introduction

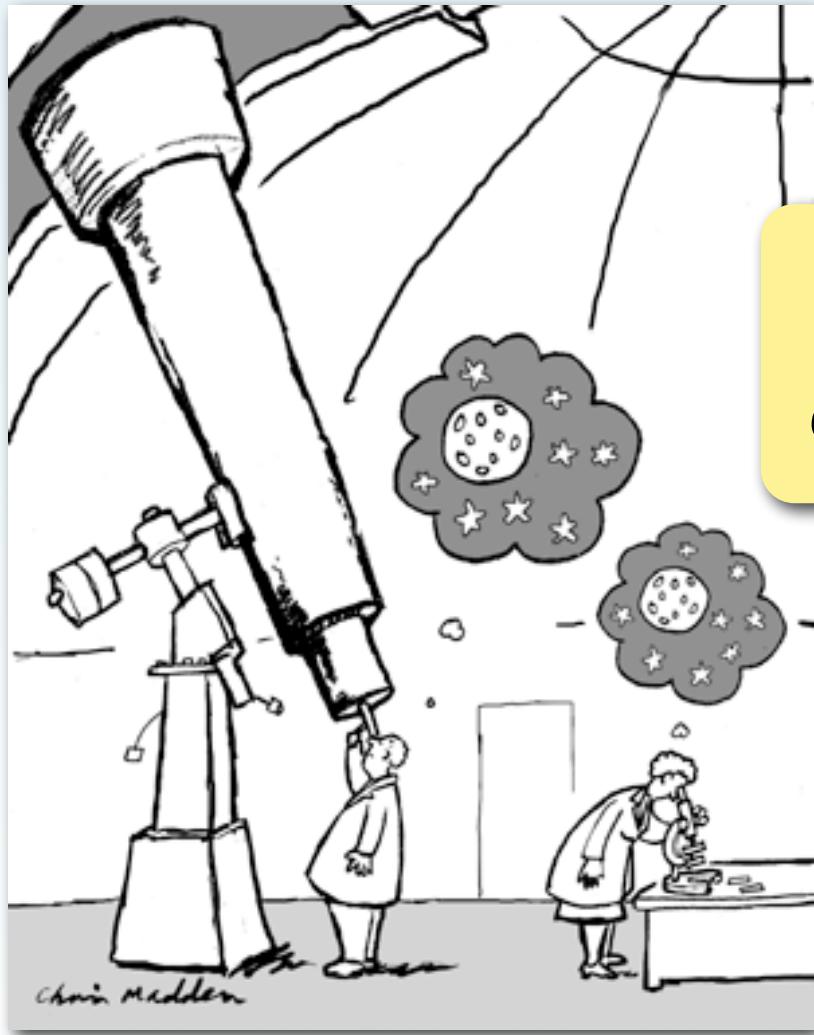
*The correct use of
language can be critical*



I Introduction

- About computer science
- The role of programming languages
- How to study programming languages?

What is Computer Science?

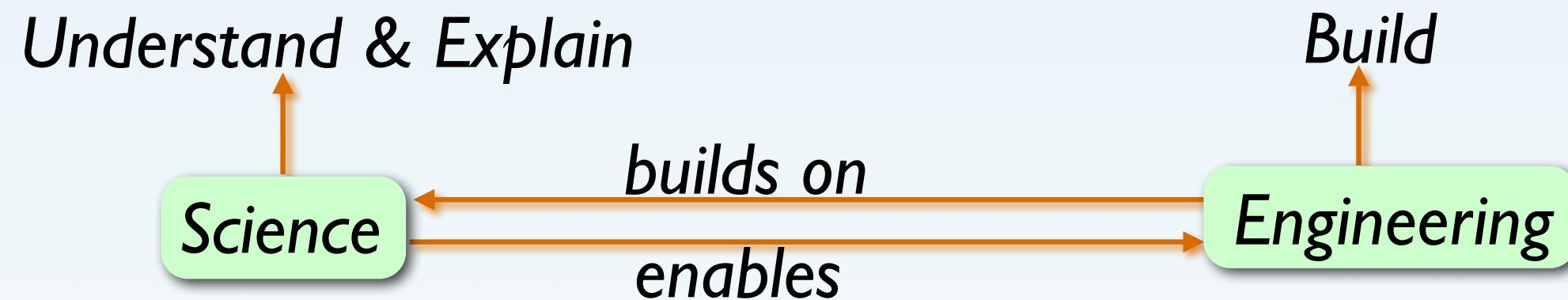


“Computer science is no more about computers than astronomy is about telescopes.”

Edsger Dijkstra (?)

Computer Science = The **Science** of Computing

Science and Engineering



*Physics
Chemistry
Theoretical CS*

*ME, EE, ...
CE, ...
SE, ...*

What is Computation?

Systematic Transformation of Representation

*Systematic
Intensional
Description*

*Transformation
Function*

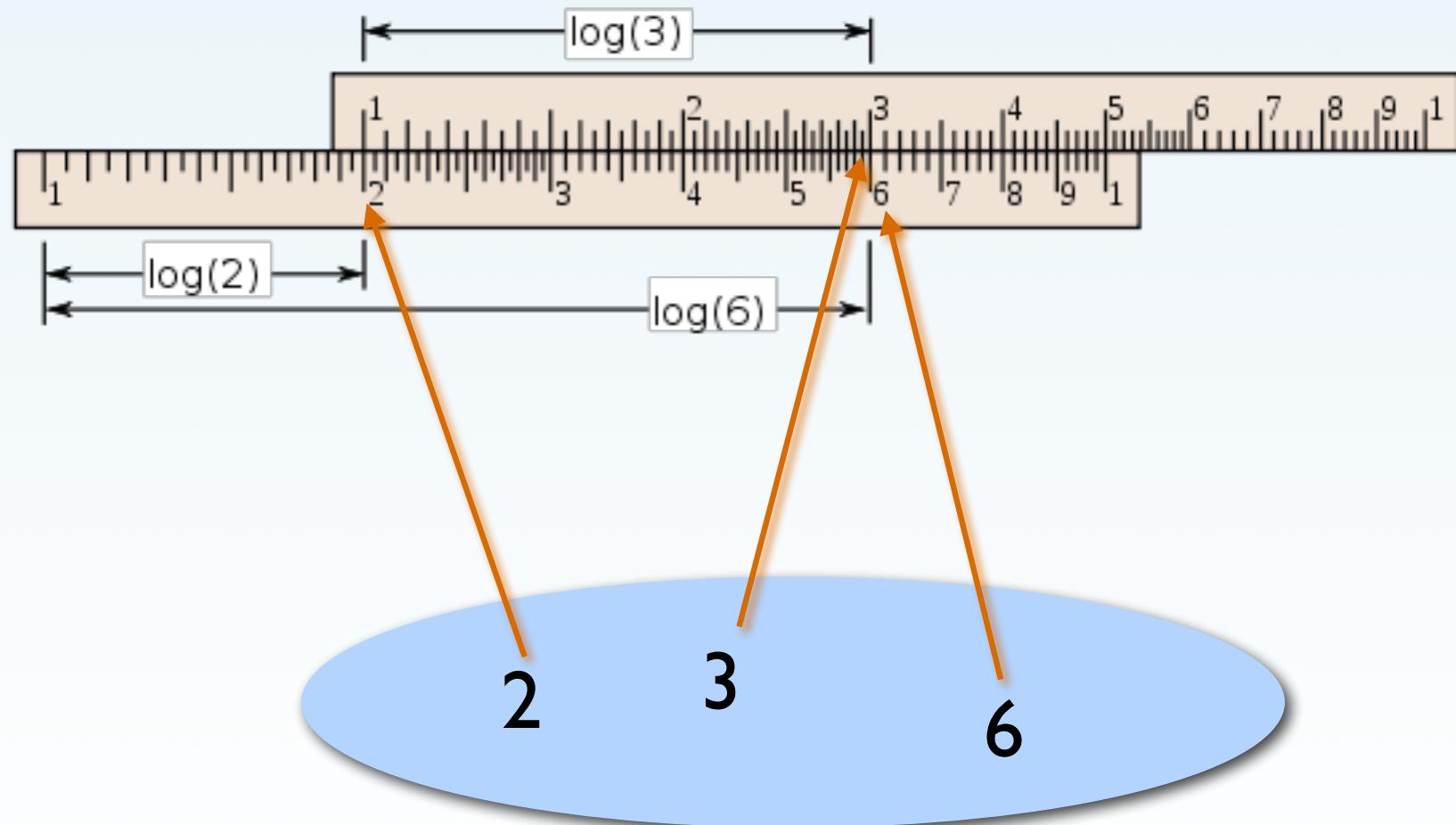
*Representation
Abstraction that preserves
particular features*

*Description given
in a Language*



Multiplication

$$\log(x \cdot y) = \log x + \log y$$



Scale alignment

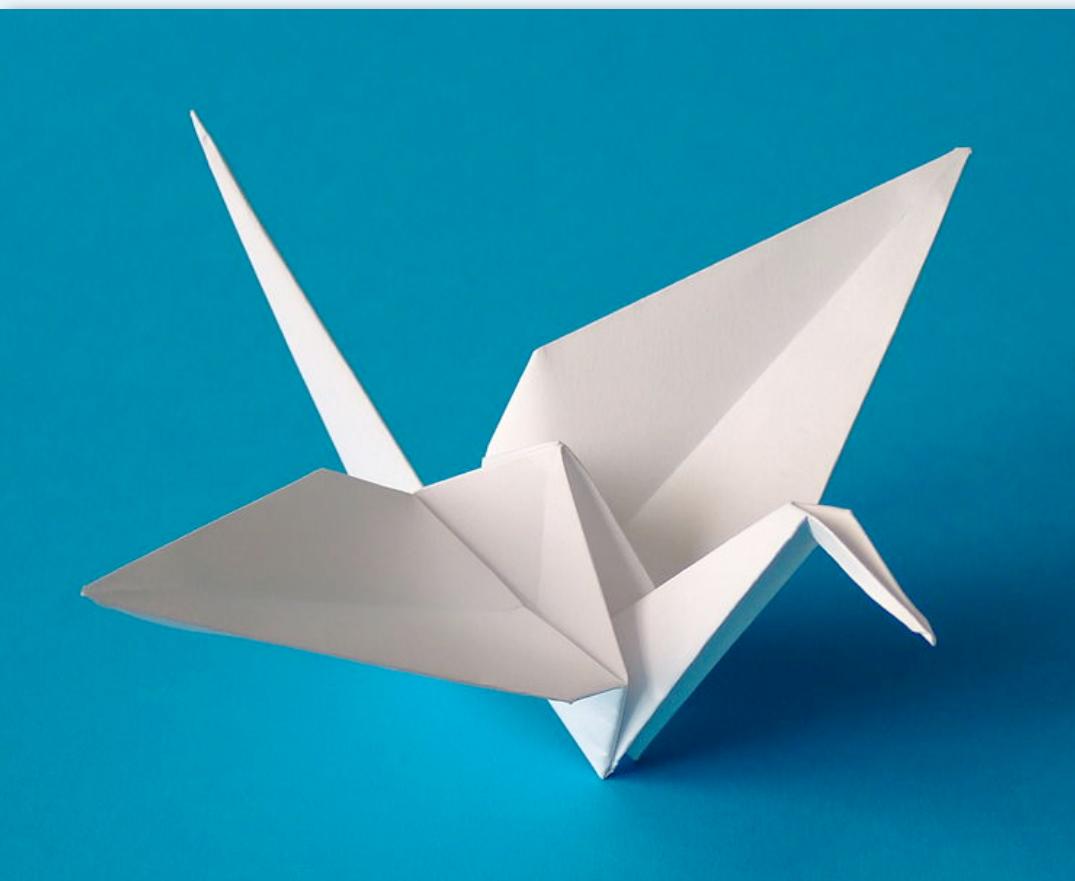
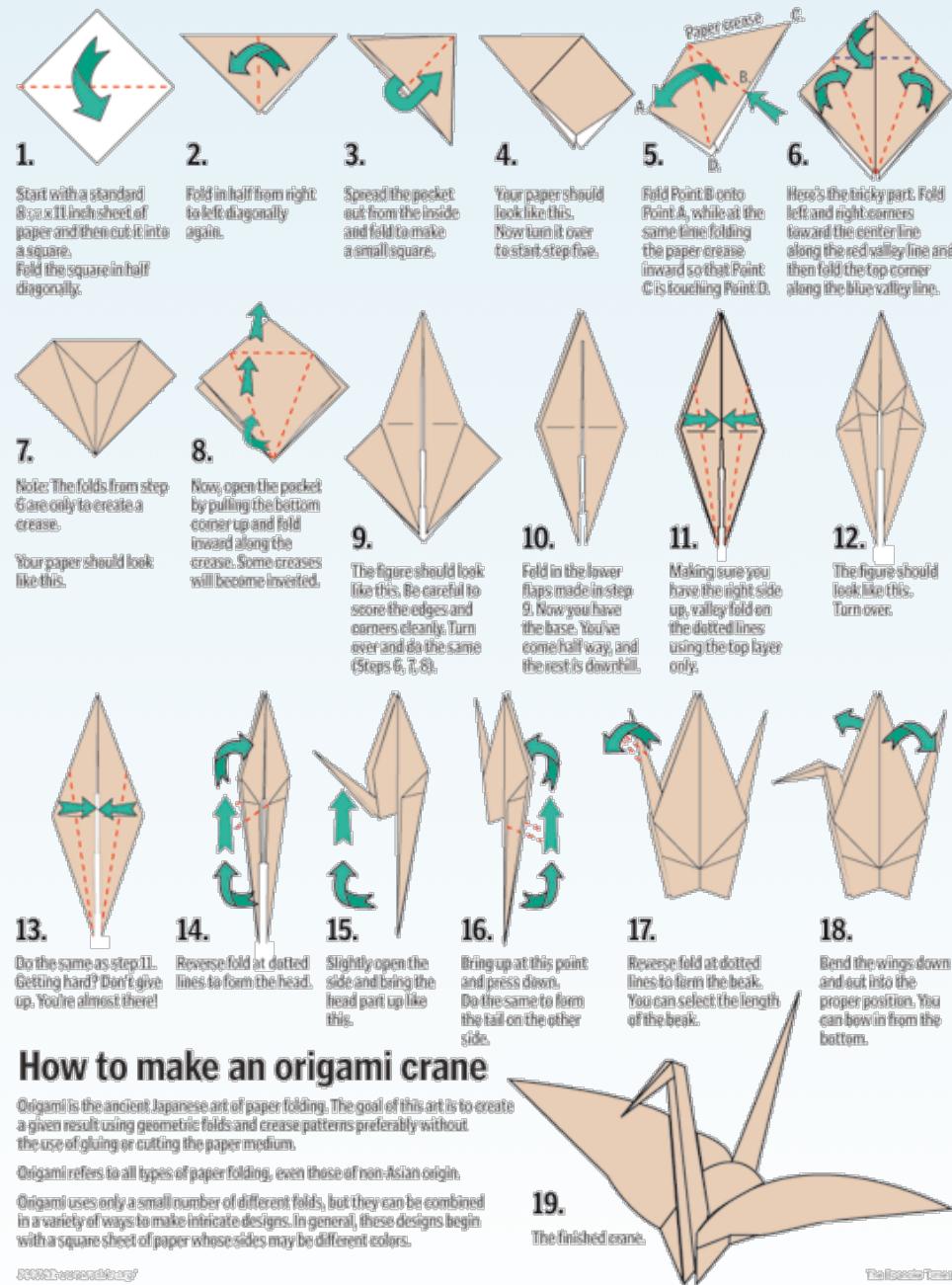
Transformation

Logarithmic Scales

Representation

Numbers

Origami



Paper folding

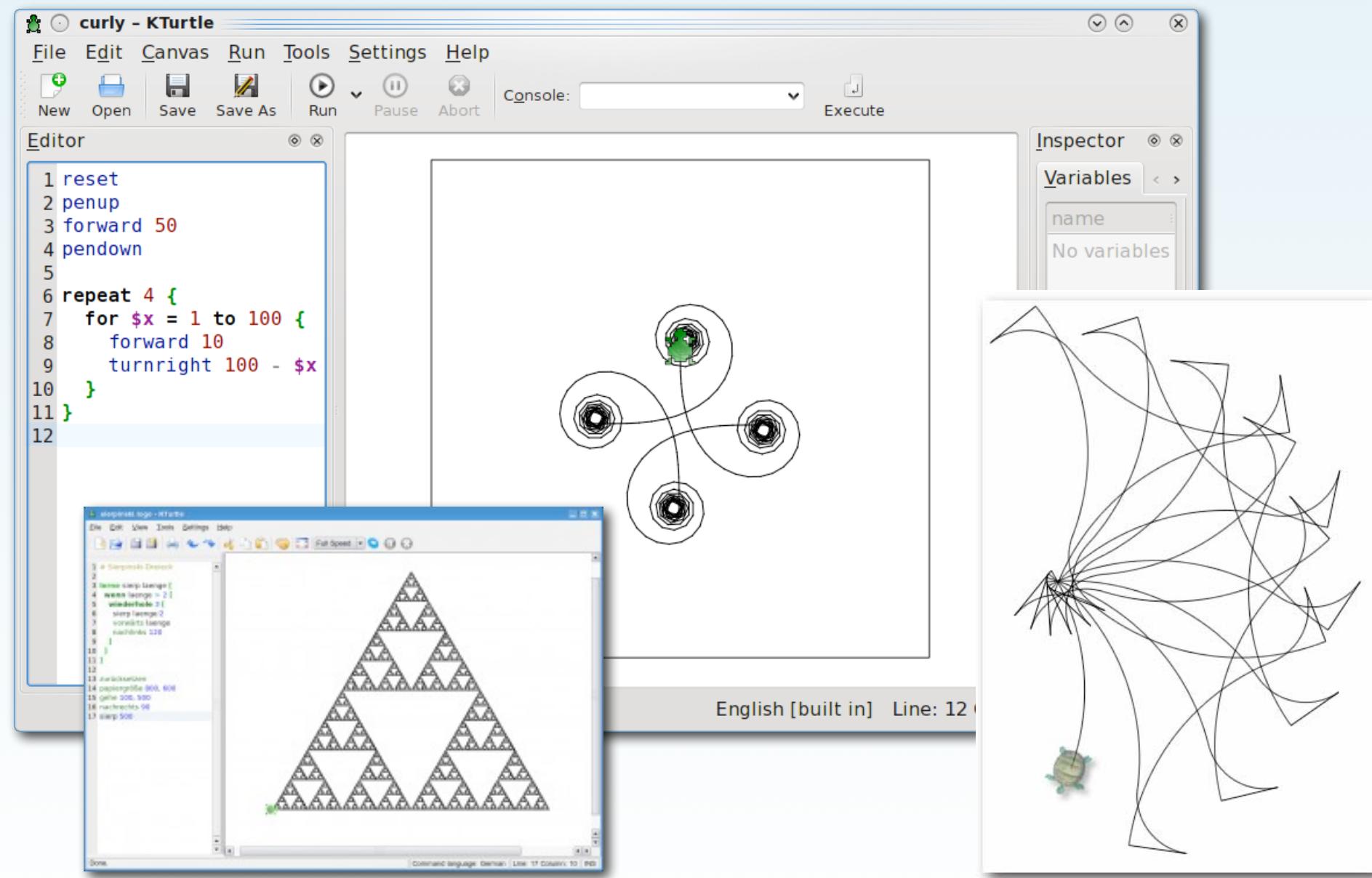
Transformation

Paper

Representation

Objects

Logo



Moving a “Turtle”

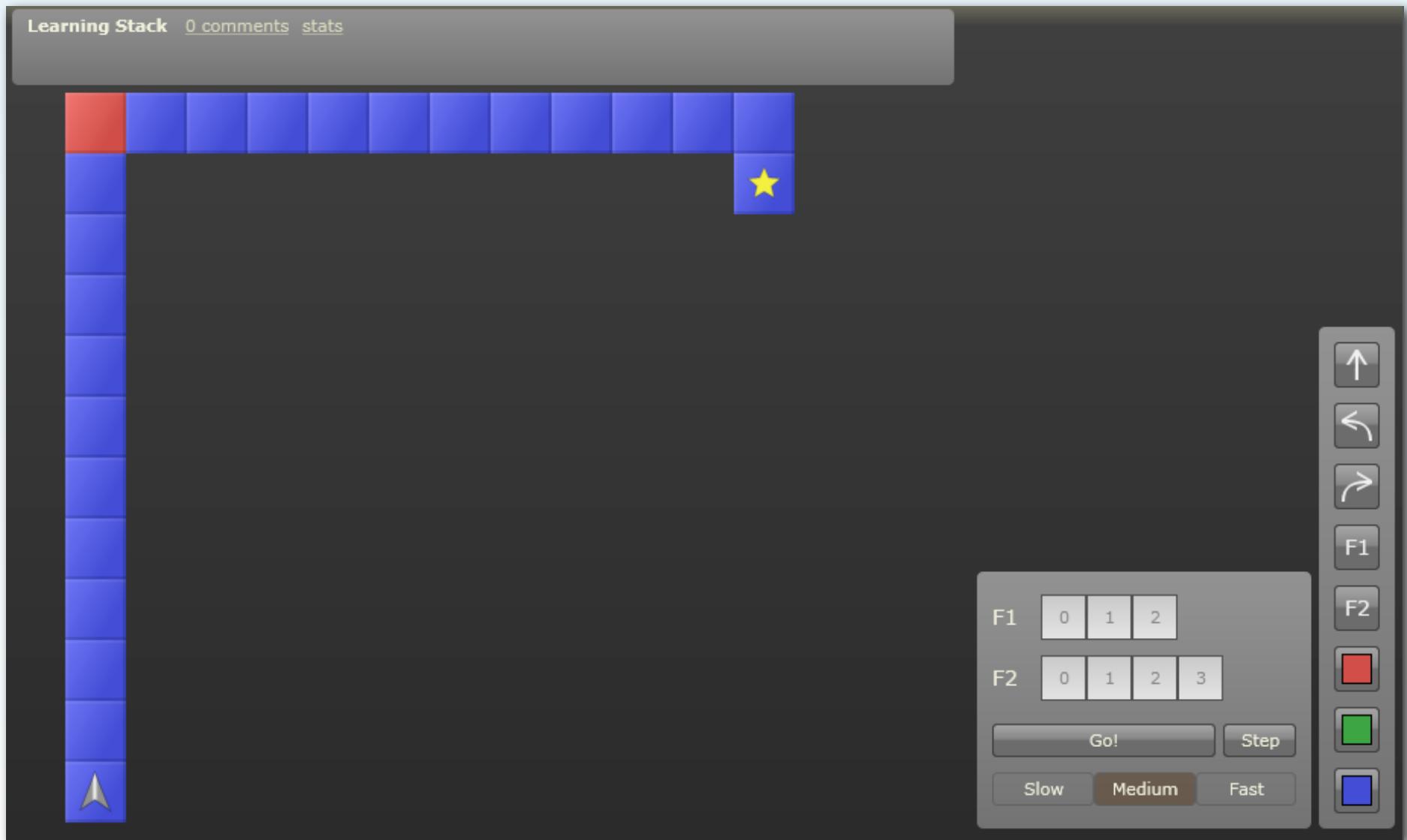
Transformation

2D Graphics

Representation

Anything visualizable

Robozzle



Moving a Robot

Transformation

Robot on a Grid

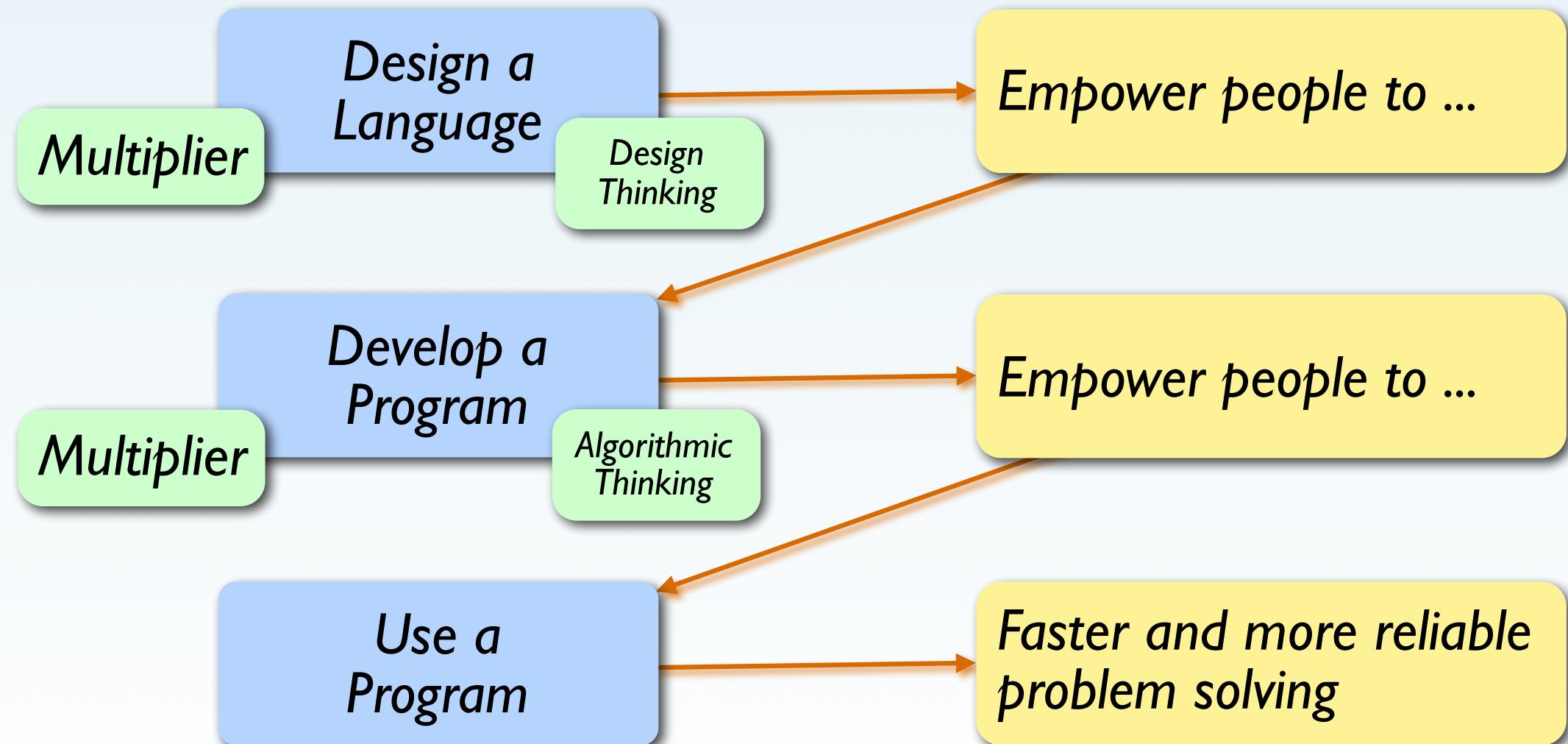
Representation

Navigation Puzzle

I Introduction

- About “computer science”
- The role of programming languages
- How to study programming languages?

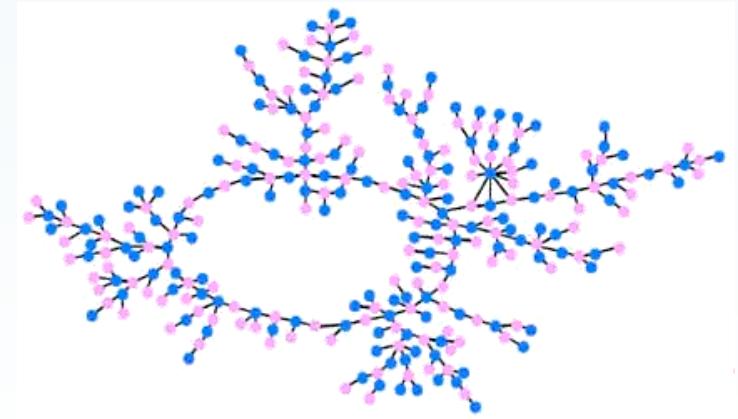
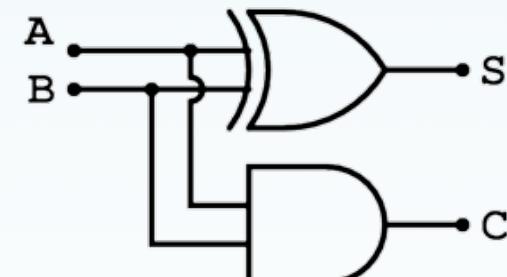
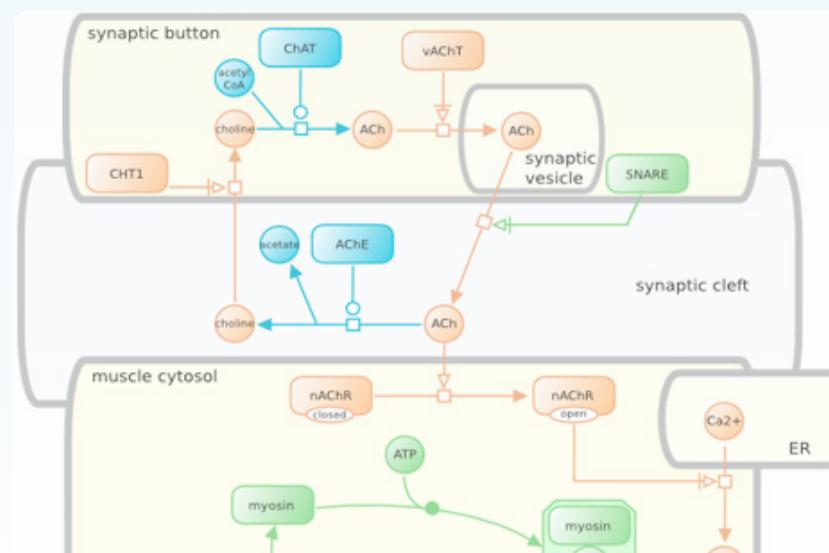
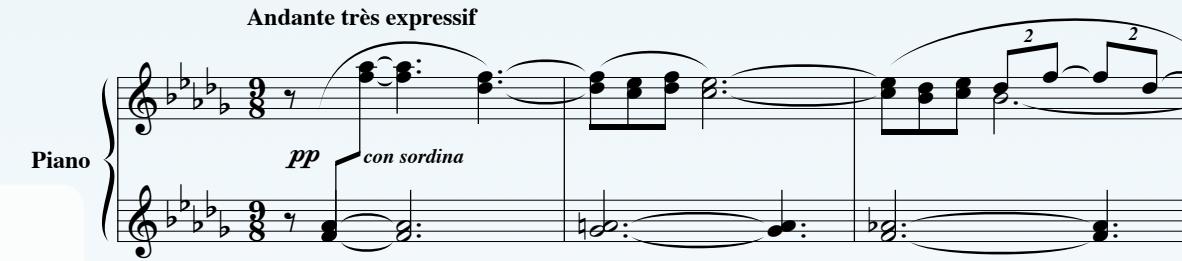
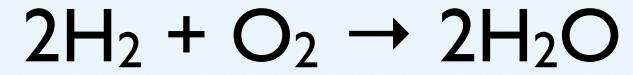
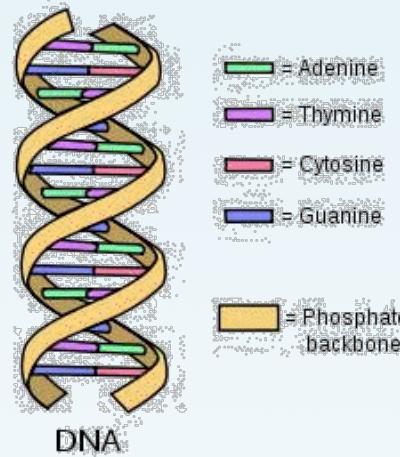
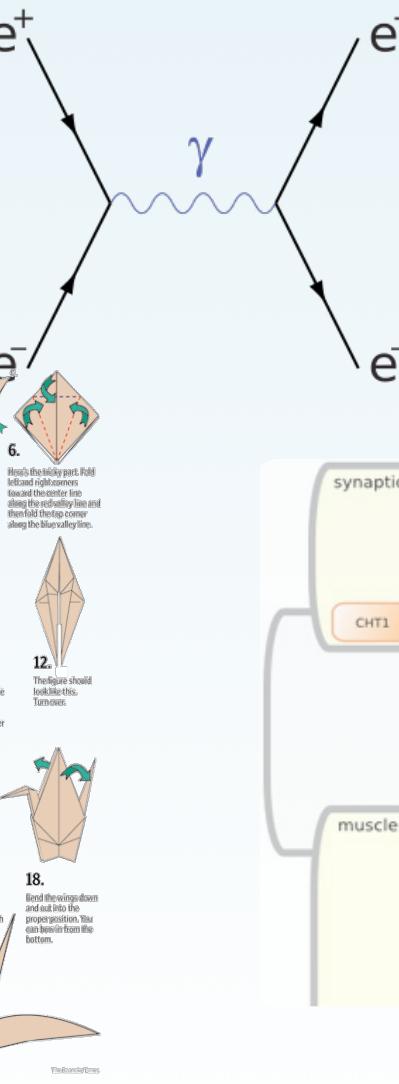
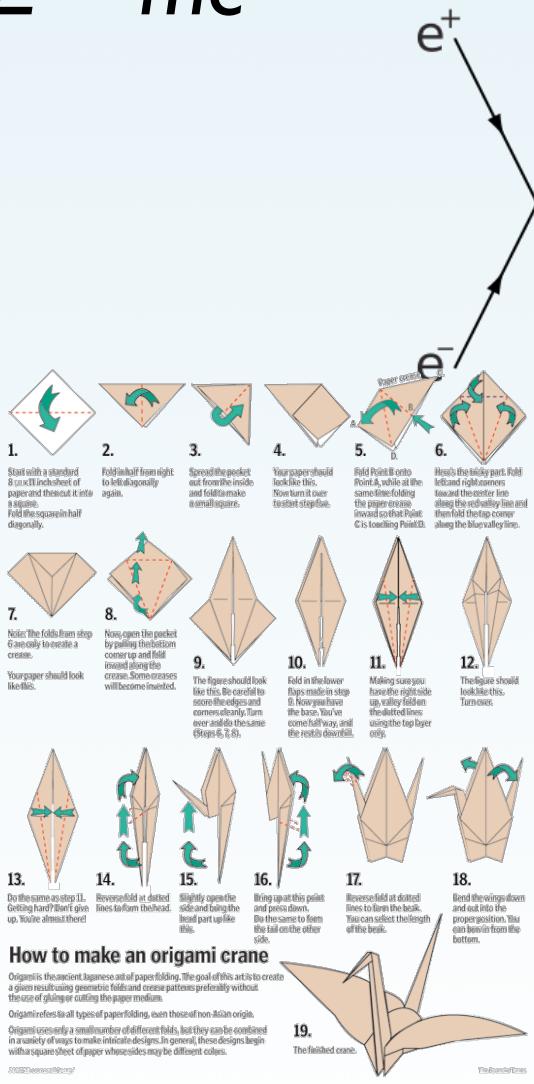
Impact of Languages & Programs



Domain-Specific Languages

$$F = ma$$

$$E = mc^2$$



DSLs vs. Programming Languages

Program: Description of a computation

⇒ A *programming language defines a range of computations*

Programming Language: Set of programs

complete: Java, C, Haskell, ...

partial: *Domain-specific languages*

(SQL, Excel, make, LaTeX, HTML, LabView, ...)

The Reach of Language Design

Languages & Language design:

- Integral part of most CS activities
- Huge impact on expressiveness & usability
- Multiplier & enabler of expressiveness
- Many applications far beyond CS

Language Design
is vital to CS

I Introduction

- About “computer science”
- The role of programming languages
- How to study programming languages?

Approach & Tools

Elements of programming language definition & understanding:

- Define *abstract syntax*
- Define *semantics*
 - *Scoping*
 - *Parameter Passing*
 - *Exceptions*
- Define *type system* (not always)
- Distinguish *paradigms*

Metalanguage

Example languages

Haskell

Prolog

Examples

Learning Objectives

	Class Section								
	1 Haskell	2 Abstract Syntax	3 Semantics	4 Types	5 Scope	6 Parameter Passing	7 Exceptions	8 Paradigms	9 Prolog
1 Abstract Syntax		✓	✓	✓					
2 Static & Dynamic Scoping					✓	✓			
3 Static & Dynamic Typing				✓				✓	
4 Parameter Passing						✓		✓	
5 Runtime Stack					✓	✓	✓		
6 Polymorphism				✓			✓	✓	
7 Exception Handling							✓		
8 Paradigms	✓	✓	✓			✓		✓	✓
9 Semantics			✓	✓	✓	✓	✓	✓	