Introduction to programing

What does a computer do?

- Performs **computation**
  - A billion calculation per second!
- **Remembers** results
  - 100s of GB of storage!

What kinds of calculation?

- Built-into the languages
- Ones that you define as the programmer

Types of knowledge?

- **Declarative knowledge** $\rightarrow$ *that* something is the case (very general)
    - J is the tenth letter of the alphabet
    - Paris is the capital of France
    - Windows OS is stupid!
    - Brain is used for thinking
- **Imperative Knowledge** $\rightarrow$ *how* to do something (Job dependent)
    - Recipe
    - factorial(n) = n * factorial(n-1)

A numerical example

Square root of a number x is y if $y * y \approx x$.

- Recipe
  1. Start with a **guess**, g
  2. if $g * g$ **close enough** to x, stop and say g is the answer.
  3. Otherwise make a **new guess** by averaging g and x/g
  4. Using the new g, **repeat** process until close enough to x

g	$g * g$	x/g	$(g + x/g)/2$
3.00000	9.00000	1.33333	2.16667
2.16667	4.69444	1.84615	2.00641
2.00641	4.02568	1.99361	2.00001
2.00001	4.00004	1.99999	2.00000

Recipe == Flow Charts

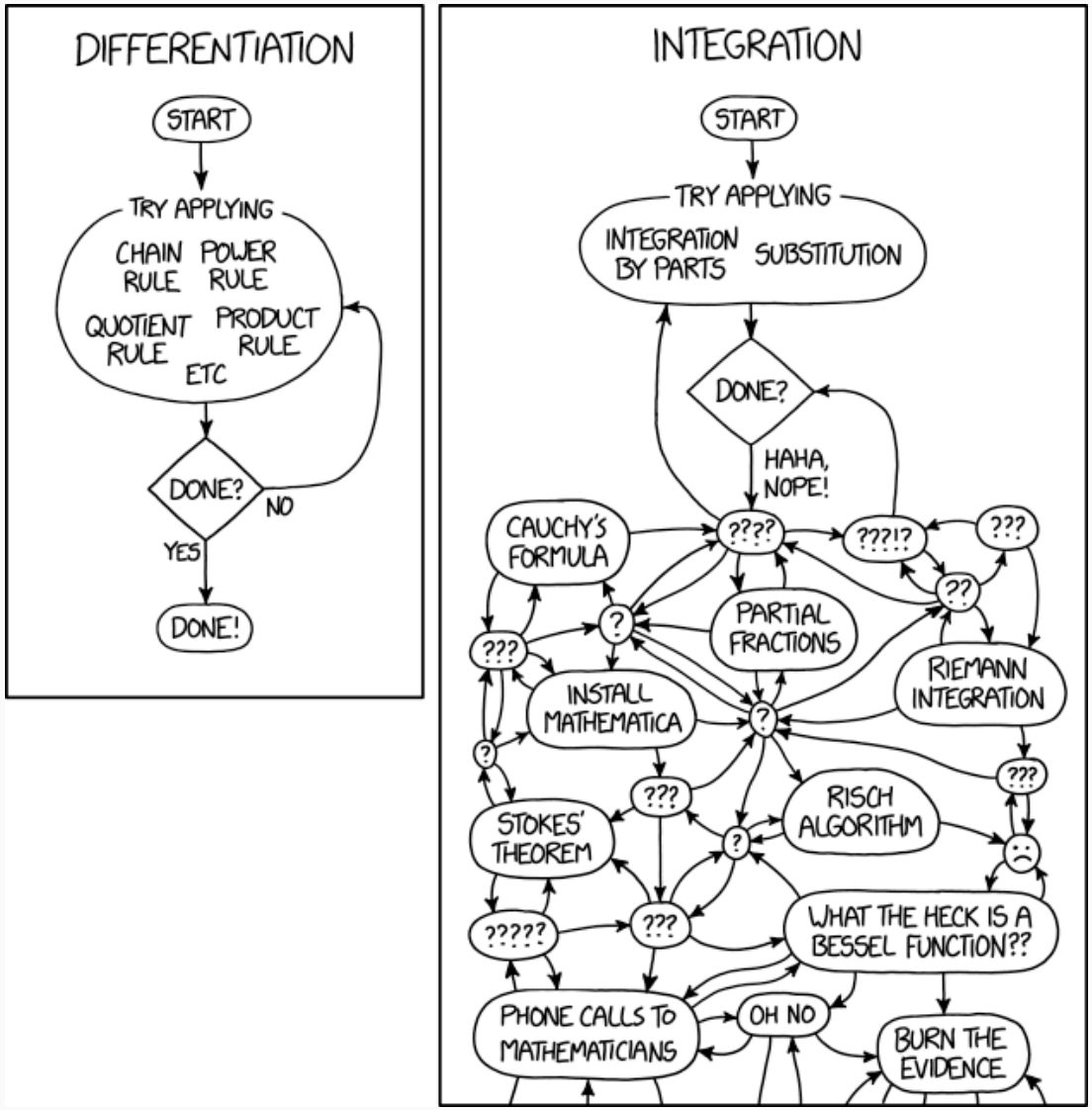
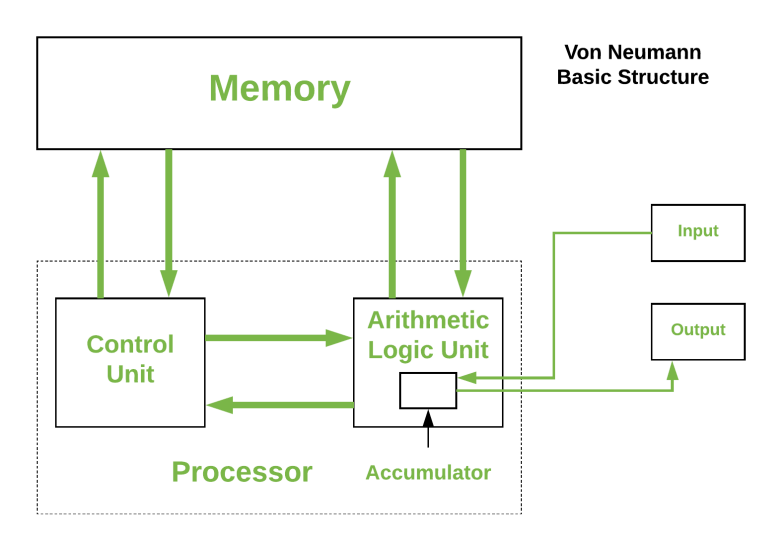
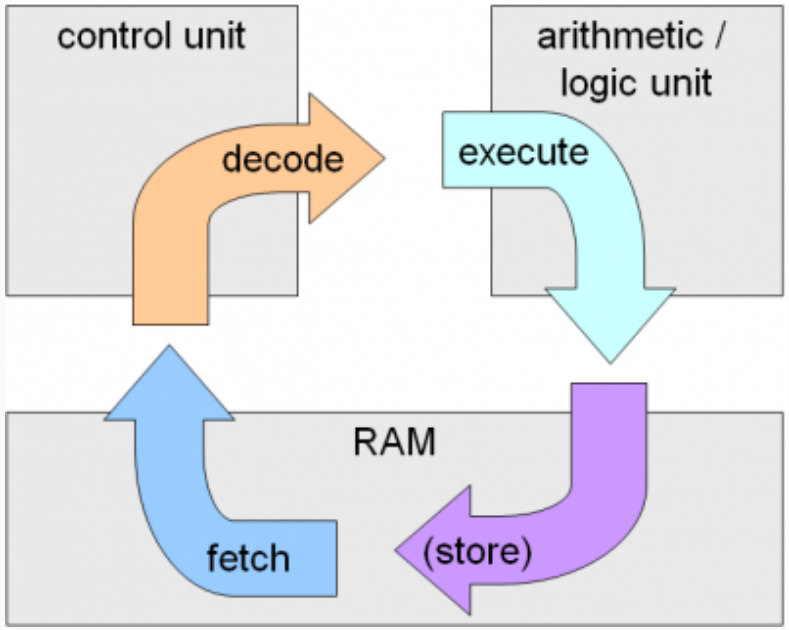


Figure 1: XKCD

Computers are machines





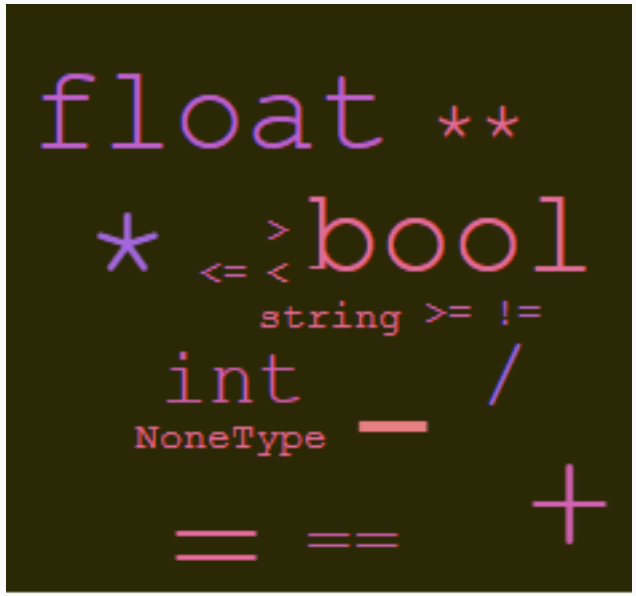
Programming languages

- Turing showed that 6 primitives are enough to computer anything!
- Modern languages has more convenient set of primitives e.g., `if then else`, `for`, `while`, `&&`, `~` etc.
- can abstract methods to **create new primitives**
- *anything computable in one language is computation in any other programming language*

Some things are easier in some languages.

Aspect of languages

- English: **words**
- programming languages: numbers, string, simple operators



1

[1] Image credit: Michael Twardos

Syntax and Semantics (meaning)

Syntax

- English
  - “cat dogs boy” (invalid syntax)
  - “cat hugs boy” (valid syntax)
- Programming language
  - `"hi" - 5` (invalid Python syntax)
  - `3.2 * 4` (valid python syntax)

Semantics

- Natural language e.g., English: Multiple meanings are possible. `"Flying planes can be dangerous."` (**context dependent**)
- Programming languages: have only one meaning (**context independent**) but may not be **what programmer intended** (aka bugs!).

Introduction to Python

- Objects
  - Program manipulates **data objects**.
  - Types: objects have a type.
  - Objects are
    - scalar (cannot be subdivided) e.g, `int`, `float`, `bool`, `None`
    - non-scalar e.g., `string`, `list`, `dict` etc.
  - Casting
- Expression
  - combine objects and operators $\rightarrow$ expressions.
  - `lhs <operator> lhs` e.g., `lhs = rhs`, `type(lhs) == type(rhs)`
  -
- Binding
- Functions