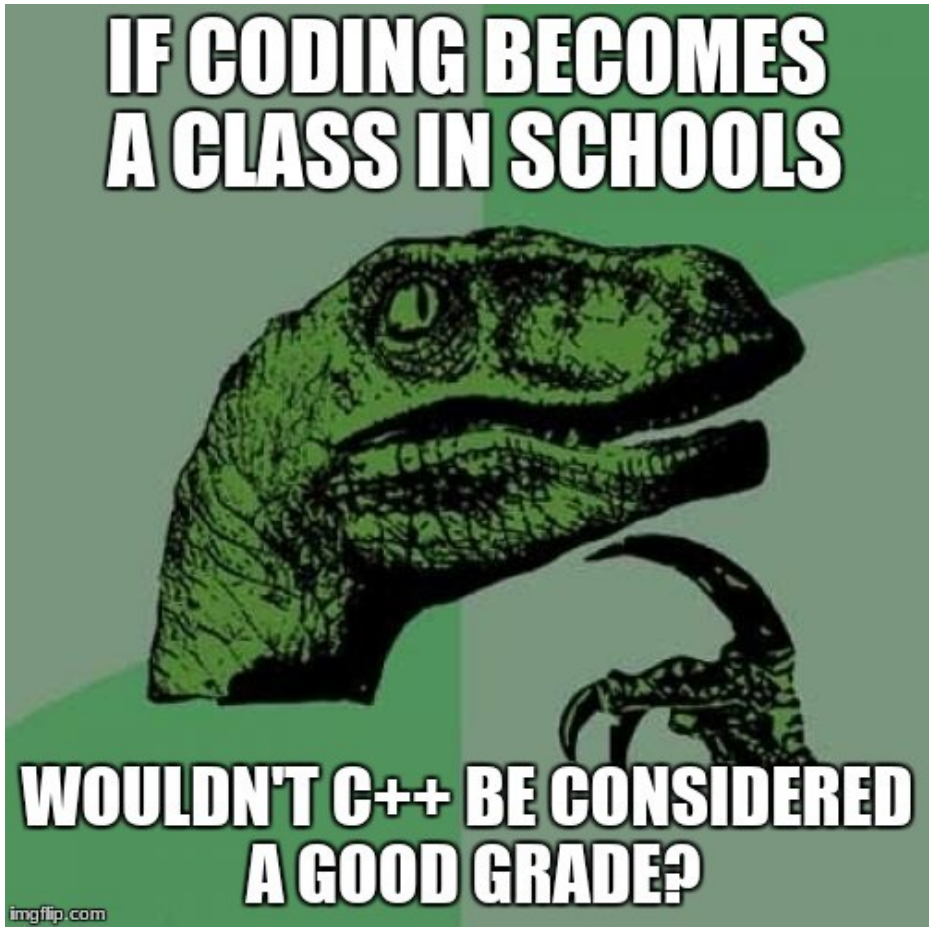

IF CODING BECOMES
A CLASS IN SCHOOLS
WOULDN'T C++ BE CONSIDERED
A GOOD GRADE?
imgflip.com

CS1428
Foundation of Computer Science

Lecture 2: What is Computing?

A super brief history of computers:

- Representing knowledge on physical media:

The Jacqard Loom
By Stephencdickson - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=79746138

- Using machines to process information:

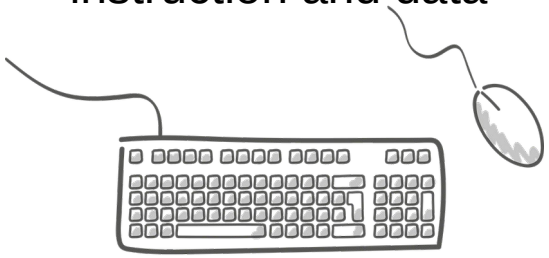
The Antikythera mechanism
CC BY 2.5, https://commons.wikimedia.org/w/index.php?curid=469865

A super brief history of computers:

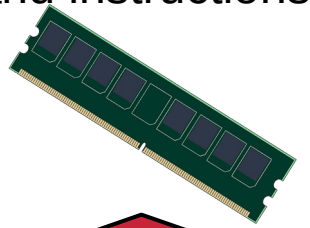
- Charles Babbage proposed (but never built) the Analytical Engine in 1837 but never built it. Ada Lovelace's correspondence with him were some of the earliest work in "computer science".
- "Colossus" was built in the UK in 1943 to support wartime code breaking.
- "ENIAC" was built in the US in 1946 to calculate artillery firing tables.
- "FLOW-MATIC", the first English-like programming language, was designed in 1955 by Grace Hopper. That language later became COBOL (which is still used)

The simplest parts of a computer:

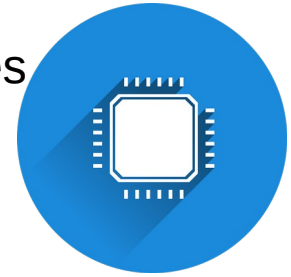
Input Devices take instruction and data

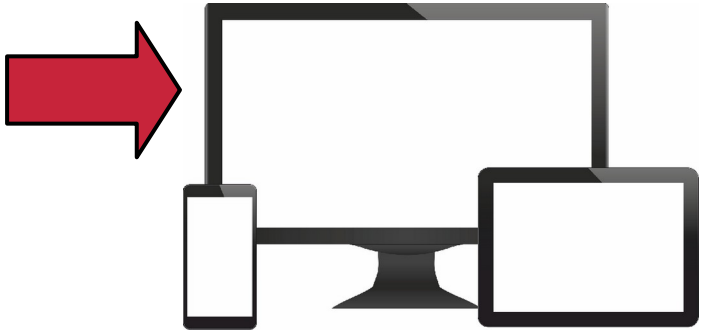


Memory stores data and instructions



A processor uses instructions to transform data



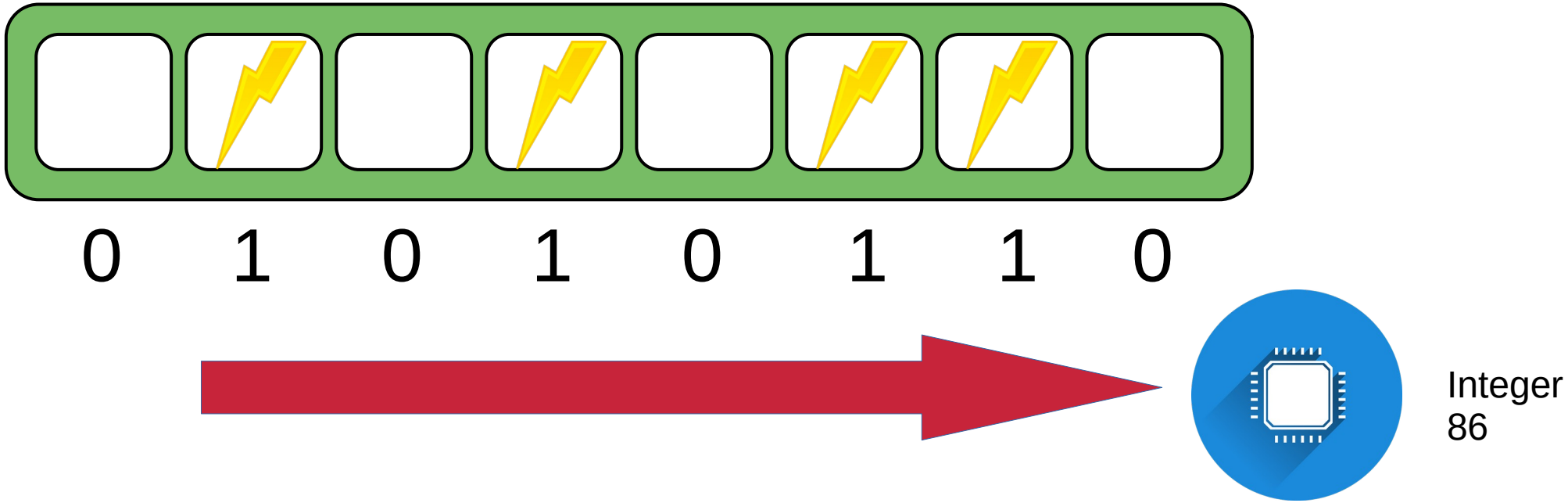


Output devices return information to the user.

How does memory store information:

- Electrical switches use high or low voltages to represent the numbers 0 and 1.
- Strings of 0s and 1s are interpreted as:
  - Integers (whole numbers)
  - Floating point numbers (real numbers)
  - Characters (letters)
  - Strings (several characters in a row)
  - Instructions (tell the processor to change some other values)

How does memory store data:



Machine Language:

- Every block of memory can either be a value or an instruction. They are stored together in the same computer memory (RAM).
- The computer has a small, built-in set of operations. Each instruction in memory makes one of those operations happen.

Assembly Language:

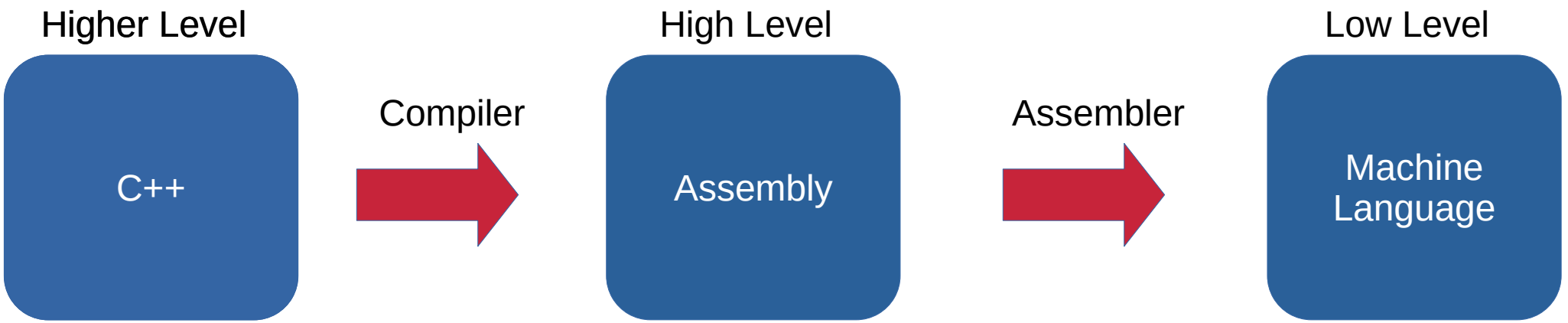
- Early computers were programmed by setting blocks of switches to 1 or 0 to represent values and machine language instructions
- Later punch cards were used to represent 1s and 0s.
- Better input/output devices let programmers write code using alphanumeric keyboards.
- Assembly language was developed to let coders write instructions in a way that looks more like English. Some example instructions:
  - LOAD- read one number out of memory
  - STORE- write one number into memory
  - ADD- sum two number and store the result somewhere

Higher Level Language:

- Assembly is easier to write than machine language but still very difficult.
- Assembly is also tied to one kind of hardware. So every computer had a unique language.
- “Higher level languages” were developed to look even more like English or algebra and are also machine independent.
- C++ is one of many higher level languages.

Translating Computer Languages:

- Assemblers turn assembly language into machine language.
- Compilers turn higher level language into assembly or higher level language into another higher level language.



Allocating memory in C++:

- “variables” are used to store some kind of value in computer memory
- Every variable has a “data type”:
  - **int-** stores a number *without* a decimal place
  - **float-** stores a number *with* a decimal place
  - **char**- stores one letter
  - **string**- stores many characters
  - **bool**- stores a true/false value

Precision of variables:

- Variables have to be stored in a certain amount of space in memory so every data type has a maximum and a minimum possible value:
  - **int-** (32 bits) -2147483648 to 2147483647
  - **float-** (32 bits) 3.4E +/- 38 (7 digits)
- The keyword **long** and **short** can be used to store integers in more or less space.
  - **long int**- (64 bits) -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
  - **short int**- (16 bits) -32768 to 32767
  - **unsigned int**- (32 bits) 0 to 4294967295
- The keyword **double** can be used to store a float in 64 bits.

Constants:

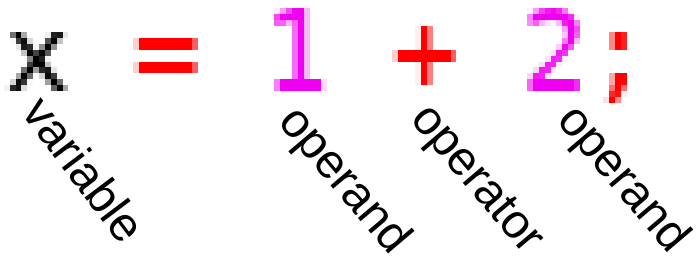
- Variables are values stored in memory that can be **changed**.
- We sometimes want to store values that **cannot** be changed.
- The **const** keyword tells the compiler that a value should be stored but never modified.
- Constants are usually named in ALL CAPS.
- **const float PI = 3.14159;**

Literals:

- Any value that you type into your program is called a literal.
- “Hi there” is a string literal.
- Literals also get stored in memory as part of your program, but cannot be referenced by name.

Modifying values:

- **Operators** can be used to change values and to store values.
- **Operators** act on **operands**
- **Operators** have to be applied in a particular order (PEMDAS)
  - ()
  - ^ * /
  - + -



Operators in C++:

- **+** add
- **-** subtract
- ***** multiply
- **/** integer or floating point division (!!!)
- **%** modulus: the remainder from integer division
- **=** assignment: fully resolves the right hand side and then stores it in the left hand side.
- **++** increment: increase the value of a variable by 1
- **--** decrement: decrease the value of a variable by 1

Integer Division:

- Think back to first learning long division in elementary school.
  - 10 / 3 = 3 remainder 1
- Floating point division:
  - 10.0 / 3.0 = 3.33333…..
- Integer division:
  - 10 / 3 = 3
  - 10 % 3 = 1
- The compiler automatically chooses integer division if **both** operands are integers and floating point division otherwise.

Combining Operators:

- **x += 1** is the same as **x = x+1**
- Operators that can be combined with assignment:
  - **+, -, *, /, %**
- Increment and decrement are also combined operators
  - **x++** is the same as **x+=1** is the same as **x=x+1**
  - **x--** is the same as **x-=1** is the same as **x=x-1**

Style Guide:

- Good style makes code easier to read, share, and maintain.
- Variable names should tell the reader how the variable is going to be used.
  - “a” is a bad variable name
  - “this_number” is a bad variable name
  - “number_of_students” is a good variable name
  - “timeOfBirth” is a good variable name
- Notice that variable can be either snake_case or camelCase (but pick one and stick with it)

Algorithms:

- An algorithm is step by step instructions to complete some task.
- A recipe is similar to the algorithm to prepare an item of food.
- The word comes from the name of 9th century mathematician Muḥammad ibn Mūsā al-Khwārizmī, which was romanticized as “Algorithmi”.
- Most code is an implementation of some algorithm.
- Never assume that the computer “knows what you mean”.