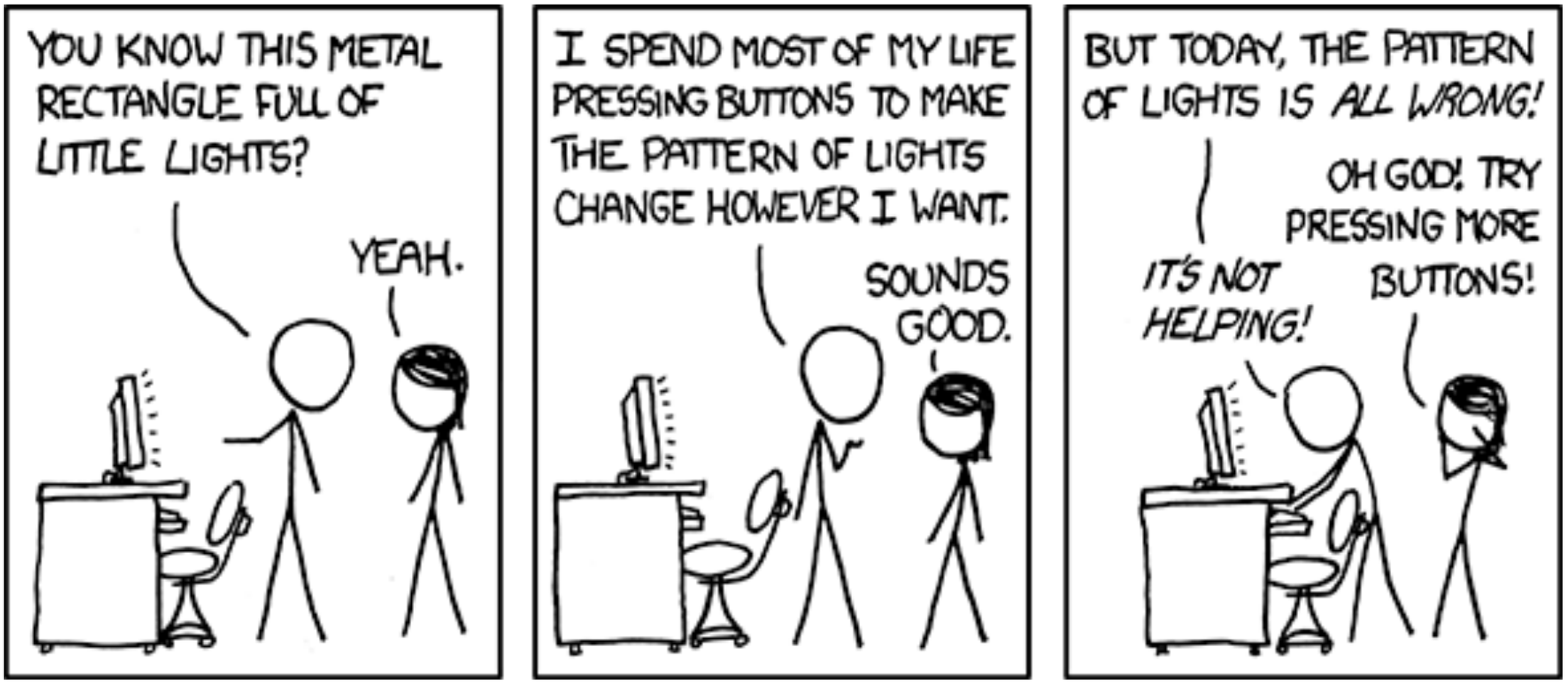


CSCI 141

Scott Wehrwein

Computers: Hardware and Software

Goals

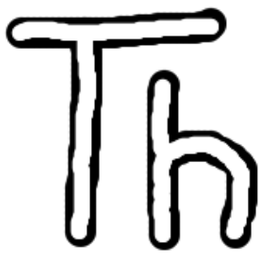
- Gain a high-level understanding of the components of a computer and how it executes programs.

Previously: Hello, world!

- **Python** is our chosen programming language in this course.

- A programming language is a language a computer can “understand” and execute (more on what this means next time)

- We’ll use a program called **Thonny** to write our Python code.



- Thonny is an example of an “Integrated Development Environment” (IDE): a program that provides all the features you need to write, run, and fix errors in programs.

Previously: Hello, world!

Our first Python program:

```
# Author: Scott Wehrwein
# Date: 3/12/2021
# Description: A program that prints
# "Hello, World!" to the screen.

print("Hello, World!")
```

Another Program

```
# Author: Scott Wehrwein
# Date: 3/12/2021
# A Python program that does a simple
# calculation and prints the result to
# the screen.

print(3*4 + 2)
```

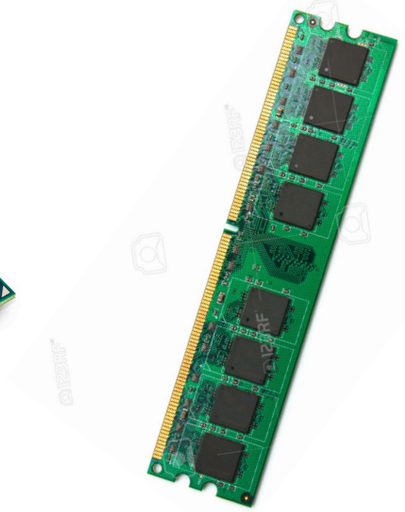
What just happened?

A lot! This course won't get into the details.

A simple model of a computer:

Input Devices

CPU



Main Memory

Secondary Storage

Output Devices

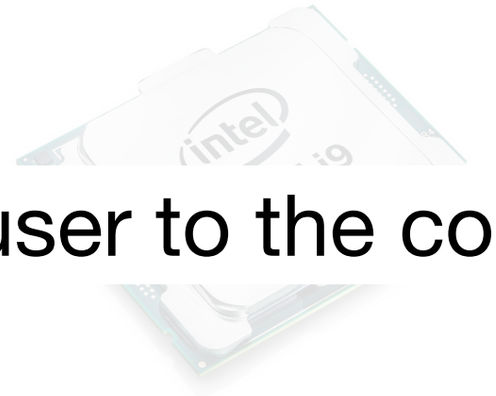
Hardware

A simple model of a computer:

Input Devices

Supply input from a user to the computer.

Output Devices



CPU

Main Memory

Secondary Storage

Hardware

A simple model of a computer:

Input Devices

CPU

Main Memory

Secondary Storage

Output Devices Transmit information back to the user.

Hardware

A simple model of a computer:

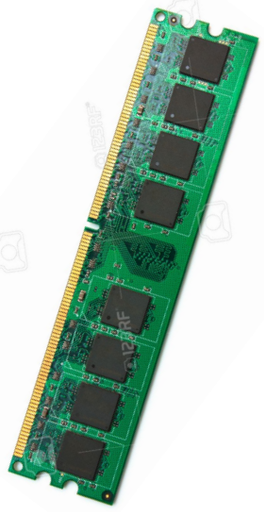
CPU:
Central Processing Unit

Executes instructions to run computer programs.

Hardware

A simple model of a computer:

Input Devices



Short-term storage:
Does not persist when the computer is turned off or the program quits.

Main Memory

also known Random Access Memory (RAM)

Secondary Storage

Output Devices

Hardware

A simple model of a computer:

Input Devices

Long-term information storage:
Stays around even if computer is off, or if program quits.

Secondary Storage

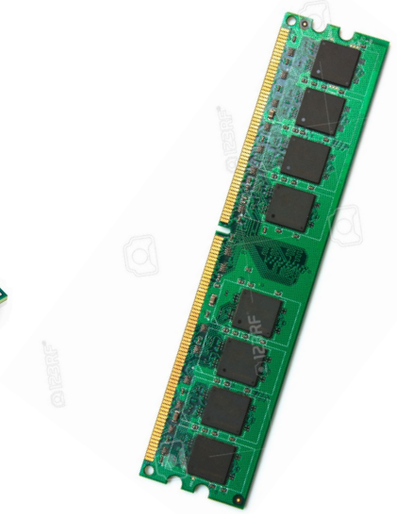
Output Devices

Hardware

A simple model of a computer:

Input Devices

CPU



Main Memory

Secondary Storage

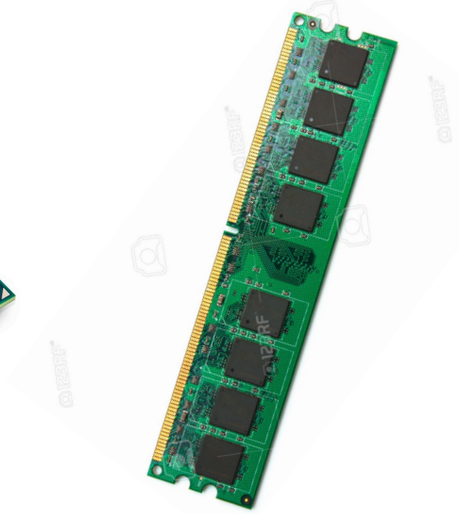
Output Devices

What can computers do?

Run programs (software).

Input Devices

CPU



Main Memory

Secondary Storage

Output Devices

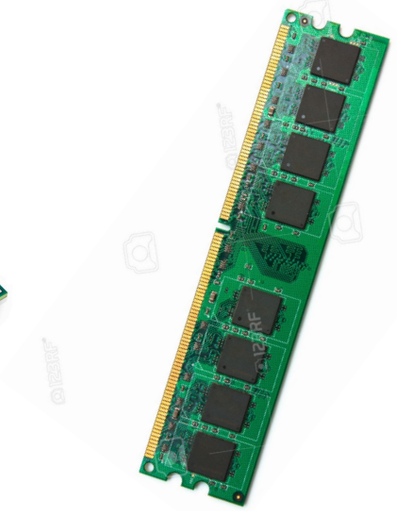
What can computers do?

Run programs (software).

That’s it!

Input Devices

CPU



Main Memory

Secondary Storage

Output Devices

How do computers run programs?

CPU

Executes instructions to run computer programs.

How do computers run programs?

Let’s take a closer look…

CPU

How do computers run programs?

Let's **not** take a closer look.

CPU

We don't need to know the hardware details!
This is an example of **abstraction**.

In brief...

- Your code is translated into simpler code
- The simpler code is translated into even simpler code
- and so on...
- ...until the instructions are so "simple"
  that an electronic circuit can do it

- The "simple" instructions are stored in **main memory**
- All the CPU does is:
  1. Fetch the next instruction from memory and "decode" it
  2. Execute it

Examples of such "simple" instructions:

- Copy a piece of data from memory into the CPU
- Do arithmetic on pieces of data in the CPU
- Copy a piece of data from the CPU to memory

How do computers run programs?

Consider a program that performs the following tasks:

- Multiply 3 by 4
- Add 2 to the result
- Print the final result to the screen.

Here are the steps that might get translated to:

- Load 3 into CPU slot A
- Load 4 into CPU slot B
- Multiply CPU slot A by CPU slot B
- Store the result in CPU slot A
- Load 2 into CPU slot B
- Add CPU slot A to slot B
- Store the result in slot A
- Print the value in slot A

The Takeaway

- Computers are made of a few important hardware components.
- A computer's job is to run software (programs).
- The CPU can run very simple instructions
- When we run python code, it is translated (automatically!) into those simple instructions.