

## Hardware Basics

You don't have to know all the details of how a computer works to be a successful programmer, but understanding the underlying principles will help you master the steps we go through to put our programs into action. It's a bit like driving a car. Knowing a little about internal combustion engines helps to explain why you have to do things like fill the gas tank, start the engine, step on the accelerator, and so on. You could learn to drive by just memorizing what to do, but a little more knowledge makes the whole process much more understandable. Let's take a moment to "look under the hood" of your computer.

Although different computers can vary significantly in specific details, at a higher level all modern digital computers are remarkably similar. Figure 1.1 shows a functional view of a computer

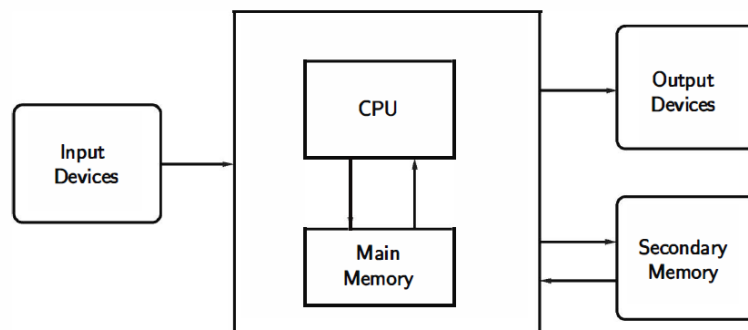


Figure 1.1: Functional view of a computer

The central processing unit (CPU) is the "brain" of the machine. This is where all the basic operations like adding two numbers and can also do logical operations like testing to see if two numbers are equal.

The memory stores programs and data. The CPU can directly access only information that is stored in main memory (called RAM for Random Access Memory). Main memory is fast, but it is also volatile. That is, when the power is turned off, the information in the memory is lost. Thus, there must also be some secondary memory that provides more permanent storage.

In a modern personal computer, the principle secondary memory is typically an internal hard disk drive (HDD) or a solid state drive (SSD). An HDD stores information as magnetic patterns on a spinning disk, while an SSD employs electronic circuits known as flash memory. Most computers also support removeable media for secondary memory such as USB memory "sticks" (also a form of flash memory) and DVDs (digital versatile discs), which store information as optical patterns that are read and written by a laser.

Human interact with the computer through input and output devices. You are probably familiar with common devices such as a keyboard, mouse and monitor (video screen). Information from input devices is processed by the CPU and may be shuffled off to the main or secondary memory. Similarly, when information needs to be displayed, the CPU sends it to one or more output devices.

So what happens when you fire up your favourite game or word processing program? First, the instructions that comprise the program are copied from the (more) permanent secondary memory into the main memory of the computer. Once the instructions are loaded, the CPU starts executing the program.

Technically the CPU follows a process called the fetch-execute cycle. The first instruction is retrieved from memory, decoded to figure out what it represents, and the appropriate action carried out. Then the next instruction is fetched, decoded, and executed. The cycle continues, instruction after instruction. This is really all the computer does from the time that you turn it on until you turn it off again: fetch, decode, execute. It doesn't seem very exciting, does it? But the computer can execute this stream of simple instructions with blazing speed, zipping through billions of instructions each second. Put enough simple instructions together in just the right way, and the computer does amazing things.

By John Zelle: Python Programming (An Introduction to Computer Science)