Introduction to ISA

1 min

An *Instruction Set Architecture*, or ISA, acts as a translator between our hardware and software. ISA is the defined set of instructions that our hardware can understand and how the software can interact with it.

Once we know what purpose the ISA is serving, we can place it into the overall hierarchy of the computer architecture. We can think of our computer system as a well-organized hamburger such as the picture on the right:

- Our top bun is the programs we interact with every day such as the web browser you are taking this class in right now.
- These programs are written in languages like Java or Python, the next layer.
- The compiler takes these languages and with the help of assembly language, translates that code into

Preview: Docs Loading link description

binary

.

- Binary code, also known as

Preview: Docs Loading link description

machine code

, conforms to the Instruction Set Architecture.

- The bottom bun is the actual hardware of the computer, the

Preview: Docs Loading link description

CPU

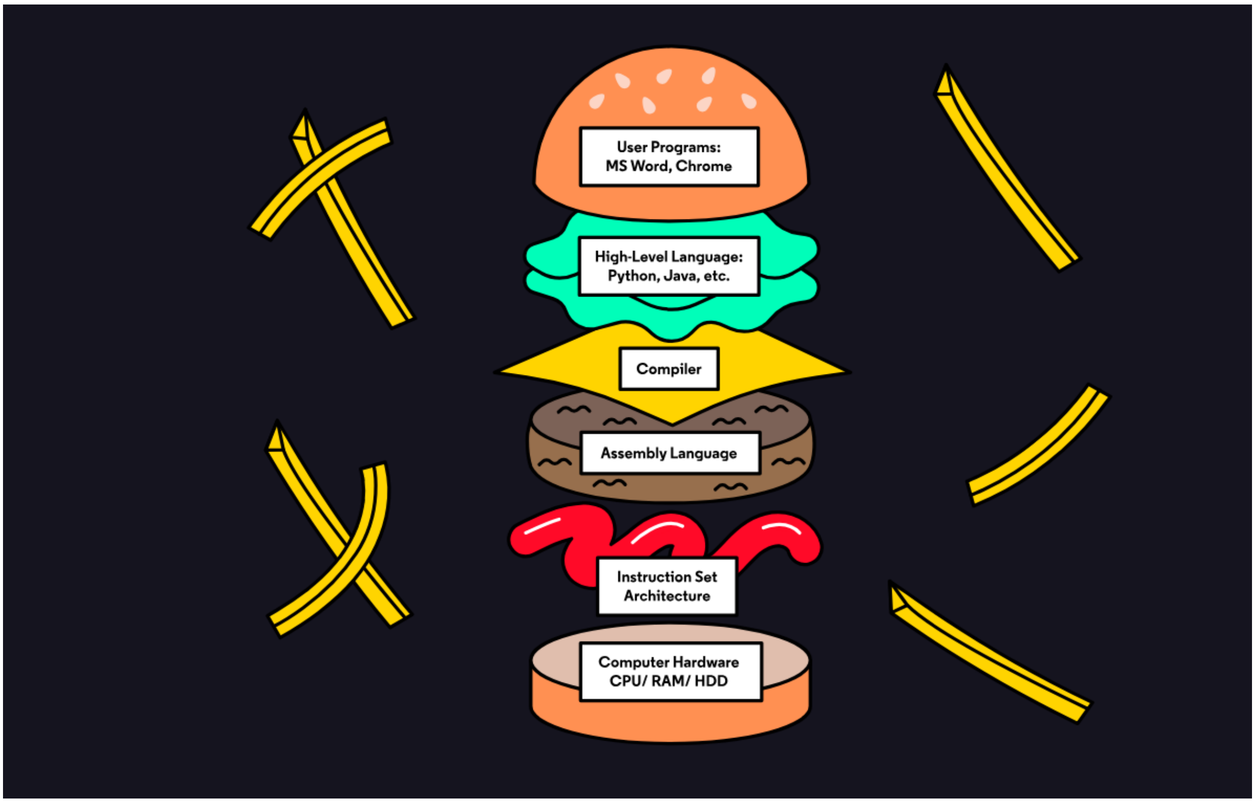
, memory, and other components, that will manipulate data based on the machine code we give it.

Unlike the other parts of our hamburger, the ISA is an abstract concept, and this can make it difficult to understand. The ISA is the agreement between the software and the hardware so that when we put in a specific sequence of binary data, the hardware will do a specific sequence of processing.

Before we can develop a good understanding of an Instruction Set Architecture, we're going to review some of the hardware components we've already learned about.

Instructions

Click Next when you're ready to go to the next exercise.


User Programs:
MS Word, Chrome
High-Level Language:
Python, Java, etc.
Compiler
Assembly Language
Instruction Set
Architecture
Computer Hardware
CPU/ RAM/ HDD