

Languages

Key Words

Languages

Translators

assembler	A program that translates assembly languages into machine code.
compiler	A program that translates high-level programming languages into machine code.
high level language	A computer programming language used to write programs in a form of English.
instruction set	The set of instructions a processor understands.
interpreter	A program that translates high-level programming languages into machine code one line at a time.
low level language	A computer programming language which closely represents machine language.
machine code	Binary representation of instruction, in a format that the CPU can process. Made up of opcode and operand.
mnemonics	Short text codes used for writing instructions in assembly code.
opcode	Operation code. Specifies instructions that can be executed by a CPU/processor in machine code.
operand	Data that is manipulated by the CPU/processor according to the given opcode
translator	Turns high-level and low-level programming languages into machine code.

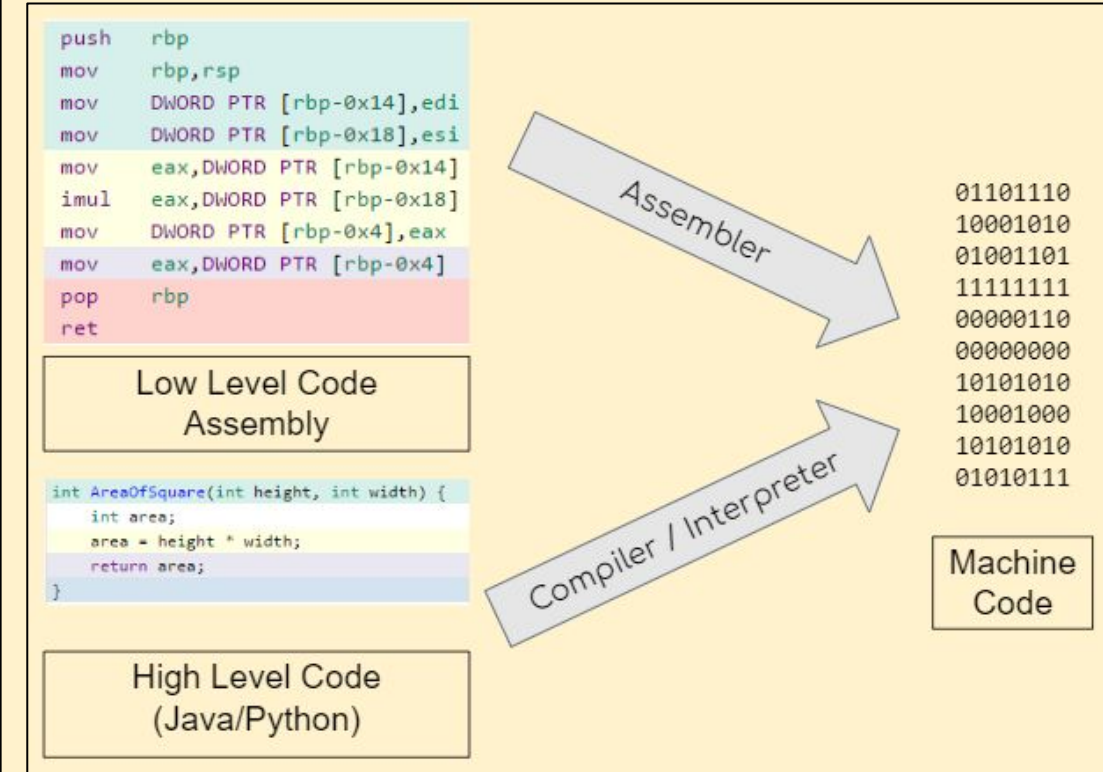
- High level languages are written like English and include Python
- Assembly Language is a low level language written using mnemonics
- Machine Code is the binary instructions that a CPU understands

01101110 10001010 01001101 11111111 00000110 00000000 10101010 10001000 10101010 01010111	<pre>push rbp mov rbp, rsp mov DWORD PTR [rbp-0x14], edi mov DWORD PTR [rbp-0x18], esi mov eax, DWORD PTR [rbp-0x14] imul eax, DWORD PTR [rbp-0x18] mov DWORD PTR [rbp-0x4], eax mov eax, DWORD PTR [rbp-0x4] pop rbp ret</pre>	<pre>int AreaOfSquare(int height, int width) { int area; area = height * width; return area; }</pre>
Machine Code	Low Level Code Assembly	High Level Code (Java/Python)

- Characteristics and purpose of different levels of programming language

	High level language	Low level language
example language	Python	Assembly
purpose	Makes writing of computer programs easier by using commands that are similar to English language.	Used for embedded systems and device drivers where instructing the hardware directly is necessary.
Instructions to machine code	One instruction translates to many machine code instructions.	One instruction translates to one machine code instruction.
type of processor	Code will run on different processors.	The code will work on one type of processor.
ease of coding	Code is quicker and easier to understand and write.	Code is much harder to understand and write.
speed of execution	Code is slower to execute.	Code is faster to execute.
memory efficiency	Less memory efficient.	More memory efficient.

- Translators are needed to translate programs written in high level languages or assembly language into the machine code that a computer understands



- Compilers and assemblers translate all the code into machine code before execution starts. An interpreter translates and executes one line at a time.

Compiler	Interpreter	Assembler
Converts the whole code into one file (often a .exe file). The file can then be run on any computer without the translator needing to be present. Can take a long time to compile source code as the translator will often have to convert the instructions into various sets of machine code as different CPUs will understand instructions with different machine code from one another.	Converts the source code into machine code one line at a time. Program therefore runs very slowly. Main reason why an interpreter is used is at the testing / development stage. Programmers can quickly identify errors and fix them. The translator must be present on the computer for the program is to be run	This type of translator is used for Assembly Language. It converts mnemonic assembly language instructions into machine code.

Quizizz

Extended Reading

Exam Questions

Video

Revision Techniques

Quizlet

Specification

