# 03_language processors



Klara Oppenheimer Schule

## Task 3: Language processors

### 3A) Read the text "language processors" and fill in the "word list".

***Language processors: Assembler, compiler, and interpreter***

Compilers and interpreters translate programs written in high-level languages into machine code that a computer understands. And assemblers translate programs written in assembly language into machine code. In the compilation process, there are several stages. To help programmers write error-free code, tools are available.

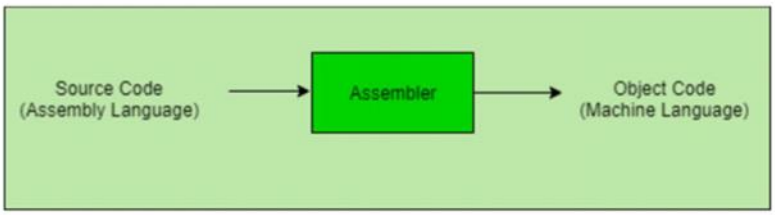
Assembly language is machine-dependent, yet mnemonics used to represent instructions in it are not directly understandable by machine and high-level language is machine-independent. A computer understands instructions in machine code, i.e. in the form of 0s and 1s. It is a tedious task to write a computer program directly in machine code. The programmes are written mostly in high-level languages like Java, C++, Python etc. and are called source code. These source codes cannot be executed directly by the computer and must be converted into machine language to be executed.

The language processors can be any of the following three types:

**Assembler**

The assembler is used to translate the program written in assembly language into machine code. The source programme is an input of an assembler that contains assembly language instructions. The output generated by the assembler is the object code or machine code understandable by the computer. Assembler is basically the 1st interface that humans need to communicate with the machine.

Code written in assembly language is some sort of mnemonics (instructions) like ADD, MUL, MUX, SUB, DIV, MOV and so on and the assembler is basically able to convert these mnemonics in binary code. Here, these mnemonics also depend upon the architecture of the machine.




**Compiler**

The language processor that reads the complete source programme written in high-level language in one go and translates it into an equivalent programme in machine language is

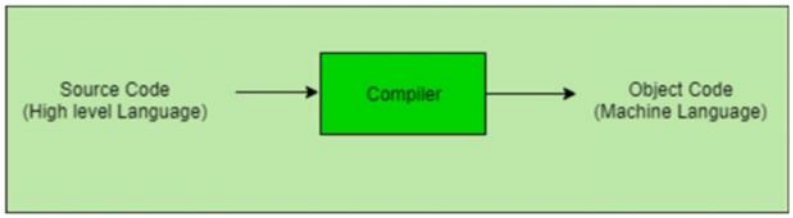
**Compiler**

The language processor that reads the complete source programme written in high-level language in one go and translates it into an equivalent programme in machine language is called a compiler.

examples: C, C++, C#.



In a compiler, the source code is translated to object code successfully if it is free of errors. The compiler specifies the errors at the end of the compilation with line numbers when there are any. The errors must be removed before the compiler can successfully recompile the source code again. The object programme can be executed any number of times without translating it again.

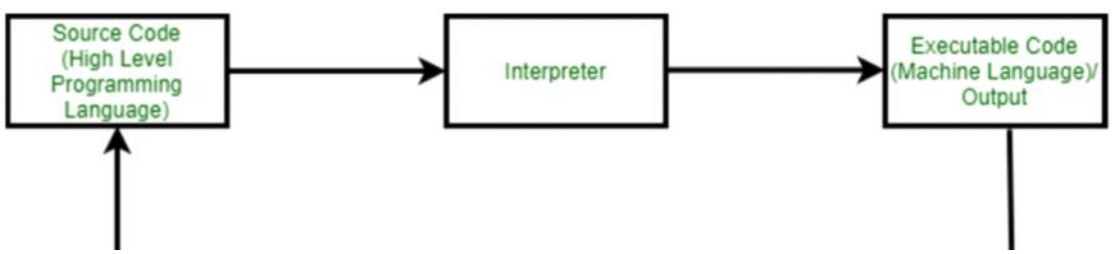


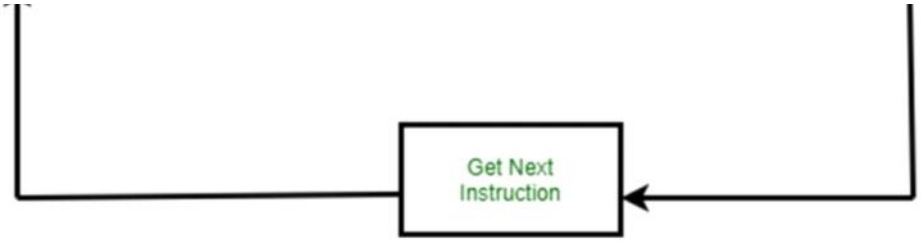

**Interpreter**

An interpreter translates a single statement of the source programme into machine code, executes it immediately, and then moves on to the next line. If there is an error in the statement, the interpreter terminates its translating process at that statement and displays an error message. The interpreter moves on to the next line for execution only after the removal of the error.

An interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code.

examples: Perl, Python, Matlab






https://www.geeksforgeeks.org/language-processors-assembler-compiler-and-interpreter/?ref=lbp (10.09.23; adapted)

Englisch 11. Klasse Computer languages

Klara Oppenheimer Schule

***Word list***

| English | German |
|---|---|
| ▪ to assemble sth | etwas zusammenfügen |
| ▪ mnemonic | Eselsbrücke |
| ▪ to convert sth into sth | konvertieren/ umwandeln |
| ▪ machine-dependent | maschinenabhängig |
| ▪ tedious | mühsam |
| ▪ to compile sth | etwas zusammenstellen (übersetzen (technisch)) |

**3B) There are three types of language processors: assembler, compiler, and interpreter. What are the differences? Write a GERMAN summary.**

### 3B) There are three types of language processors: assembler, compiler, and interpreter. What are the differences? Write a GERMAN summary.

Ein Assembler übersetzt ein Programm aus einer Assembly-Sprache in Maschinen-sprache (Binärcode).

Menschen können kaum in Maschinensprache kommunizieren, deshalb benutzen sie eine Assembly-Sprache, die bestimmte Zeichen enthält(ADD,SUB, usw.)

Ein Compiler überträgt den Quellcode komplett in Maschinensprache, das Programm kann dann sofort und immer wieder ausgeführt werden. Wenn beim Übertragen ein Fehler passiert, kann der Vorgang nicht fortgesetzt werden, bis der Fehler behoben ist. Der Compiler zeigt an, in welchen Zeilen sich Fehler befinden.

Ein Interpreter überträgt den Quellcode Anweisung für Anweisung in Maschinensprache, führt die Befehle sofort aus und fählrt dann mit der nächsten Zeile fort. Bei einem Fehler wird der Vorgang abgebrochen, der Interpreter fährt erst mit der nächsten Zeile fort, wenn der Fehler behoben ist.



### 3C) Talk to a partner. What are the advantages and disadvantages of compilers and interpreters?

### Fill in the table.

| | compiler | interpreter |
|---|---|---|
| amount of time needed | takes a large amount of time to analyse the entire source code but the overall execution time of the program is comparatively shorter | takes less time to analyse the source code but the overall execution time of the program is longer |
| debugging | | |

| | | |
|---|---|---|
| debugging | the compiler generates the error message only after scanning the whole program, so debugging it is comparatively hard as the error can be present anywhere in the program | debugging is easier as it continues translating the program until the error is met |
| amount of memory needed | | |
| security | | |