

# Language Translator

---

# Language Translator

---

- A program written in high-level language is called as source code. To convert the source code into machine code, translators are needed.
- A translator takes a program written in source language as input and converts it into a program in target language as output.
- It also detects and reports the error during translation.

## **Roles of translator are:**

- Translating the high-level language program input into an equivalent machine language program.
- Providing diagnostic messages wherever the programmer violates specification of the high-level language program.

## **Different type of translators**

---

The different types of translator are as follows:

**Compiler**

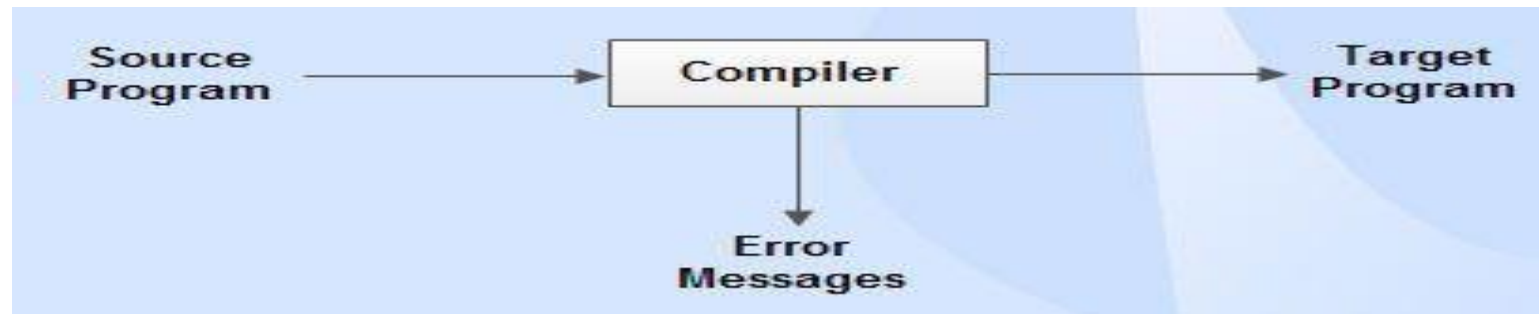
**Interpreter**

**Assembler**

---

## Compiler

Compiler is a translator which is used to convert programs in high-level language to low-level language. It translates the entire program and also reports the errors in source program encountered during the translation.



---

## Interpreter

Interpreter is a translator which is used to convert programs in high-level language to low-level language. Interpreter translates line by line and reports the error once it encountered during the translation process.

It directly executes the operations specified in the source program when the input is given by the user.

It gives b



---

## Assembler

Assembler is a translator which is used to translate the assembly language code into machine language code.



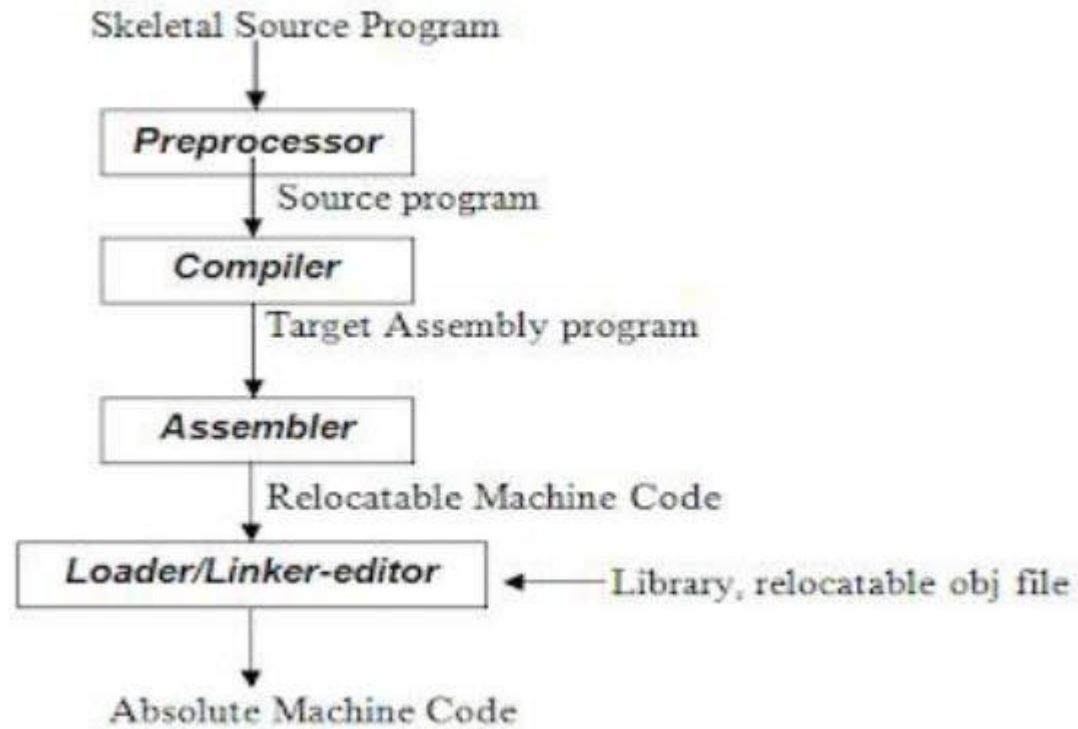
## Differences between compiler and interpreter

---

Sl. No	Compiler	Interpreter
1	Performs the translation of a program as a whole.	Performs statement by statement translation.
2	Execution is faster.	Execution is slower.
3	Requires more <b>memory</b> as linking is needed for the generated intermediate object code.	<b>Memory</b> usage is efficient as no intermediate object code is generated.
4	Debugging is hard as the error messages are generated after scanning the entire program only.	It stops translation when the first error is met. Hence, debugging is easy.
5	Programming languages like C, C++ uses compilers.	Programming languages like <b>Python</b> , BASIC, and Ruby uses interpreters.

## Structure of LanguageTranslator

---





# Preprocessor

---

A preprocessor produce input to compilers. They may perform the following functions.

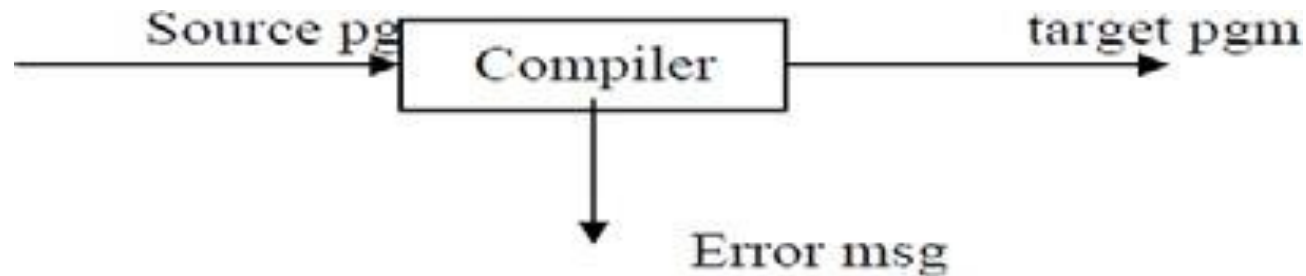
1. *Macro processing:* A preprocessor may allow a user to define macros that are short hands for longer constructs.
2. *File inclusion:* A preprocessor may include header files into the program text.
3. *Rational preprocessor:* these preprocessors augment older languages with more modern flow-of- control and data structuring facilities.
4. *Language Extensions:* These preprocessor attempts to add capabilities to the language by certain amounts to build-in macro

## Compiler

---

Compiler is a translator program that translates a program written in (HLL) the source program and translate it into an equivalent program in (MLL) the target program.

As an important part of a compiler is error showing to the programmer.



## Assembler

---

Programmers found it difficult to write or read programs in machine language.

They begin to use a mnemonic (symbols) for each machine instruction, which they would subsequently translate into machine language.

Such a mnemonic machine language is now called an assembly language.

Programs known as assembler were written to automate the translation of assembly language into machine language.

The input to an assembler program is called source program, the output is a machine language translation (object program).

## Linker

---

A linker is special program that combines the object files, generated by compiler/assembler, and other pieces of codes to originate an executable file have. exe extension.

In the object file, linker searches and append all libraries needed for execution of file.

It regulates the memory space that will hold the code from each module.

## Loader

---

“A loader is a program that places programs into memory and prepares them for execution.”

It would be more efficient if subroutines could be translated into object form the loader could “relocate” directly behind the user’s program.