1.1 Language Processors

1 What is a compiler ?

Is a program that can read a program in one language (source language)

Translate this program into equivalent program in another language (target language)

- Its important role

Is to report any errors in the source program that it detects during the translation process

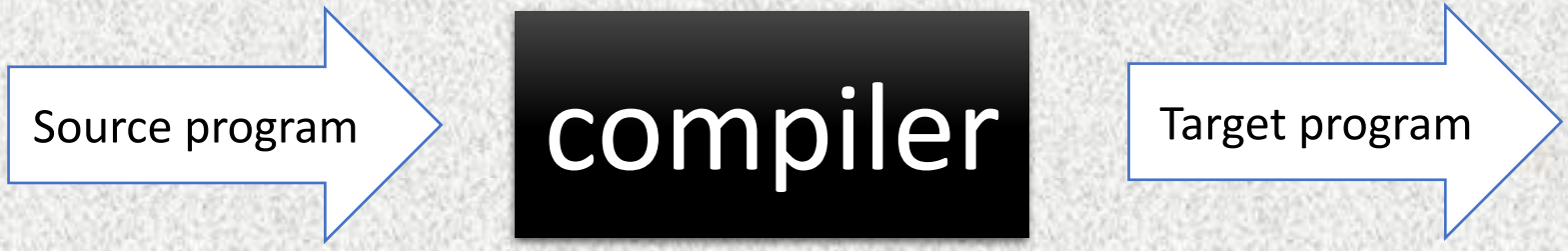

Figure 1.1 A compiler

If the target program is an executable machine-Language program it can then be called by the user to process inputs and produce outputs

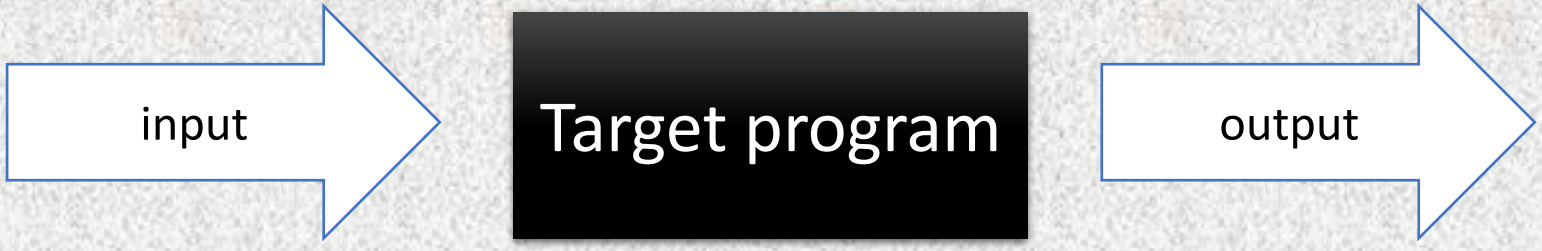

Figure 1.2 Running the target program

2 What is interpreter

It is also language processor

Instead of producing a target program as a translation , it directly execute the operations specified in the source program on inputs supplied by the user

Figure 1.3 An interpreter

3 Compiler V.S Interpreter

	Item	Compiler	Interpreter
1	Target Program	Faster the interpreter	Slower than compiler
2	error	lower diagnostics than interpreter	Better diagnostics than compiler
3	Program execution	Block of code	Step by step
4	memory	Require larger memory	Don’t require larger memory
5	Intermediate code generated	There is	There isn’t

6	Saving	Object code generated is temporary saved	Machine language code generated is never saved
7	Dynamic typing	Difficult to implement for compiler based language	Is supported by interpreter based language
8	Language Example	C and C++	Javascript , PHP and Ruby

4 Advantages : Interpreter over compiler

1. The debugging of an interpreted program is comparatively easy . As single line of code is translated and executed at a time
2. Errors are reported by the interpreter for a single line of code at a time
3. Interpreters are memory efficient as no temporary storage of the translated code takes place unlike as in a compiler where the Object code is temporarily stored on the disk
4. The Interpreter analyzes one line at a time and thus needs less time for analyzing, however; the Compiler analyzes the complete program in one go and thus needs more time for analyzing.

5 Advantages : compiler over interpreter

1. As compilers analyze the program before compiling it, this ensures all errors are identified and corrected before the compiled code is generated.

2. An intermediate code (Object code generated) which can then be used each time the program is to be run, thus eliminating the need for compiling the source program each time

3. Compiling a program is usually faster than interpreting it.

4. The Interpreter analyzes one line at a time and thus needs less time for analyzing, however; the Compiler analyzes the complete program in one go and thus needs more time for analyzing.

6 Frequently Asked Questions

Q #1) What is the difference between compiler vs interpreter?

Compilers and Interpreters perform the same job of translating a High-level program to Machine language In the case of a Compiler, the entire program is converted to machine code in one go whereas in the case of an Interpreter the translation happens one line of code at a time

Q #2) What are a compiler?

Compilers are translation programs that convert High-level program code to machine language code so that it can be understood by the computer

Q #3) What are the three types of translators ?

- Compilers
- Interpreters
- Assemblers

Q #4) What is the difference between compiler and interpreter and assembler?

Compiler: This is a translation program that converts a complete high-level program code to machine code in one go.
Interpreter: This is also a translation program that converts a complete high-level program code to machine code but one line of code at a time.
Assembler: This is a translation program that converts a code written in Assembly language to machine code.