Compilation and Interpretation

Introduction

There are two primary methods for translating high-level code:

- Compilation
- Interpretation

Compilation

Compilers were the first sort of translator program to be written. The idea is simple: You write the program, then hand it to the compiler which translates it. Then you run the result.

Compiler characteristics:

- Spends a lot of time analyzing and processing the program
- The resulting executable is some form of machine- specific binary code
- The computer hardware interprets (executes) the resulting code
- Program execution is fast
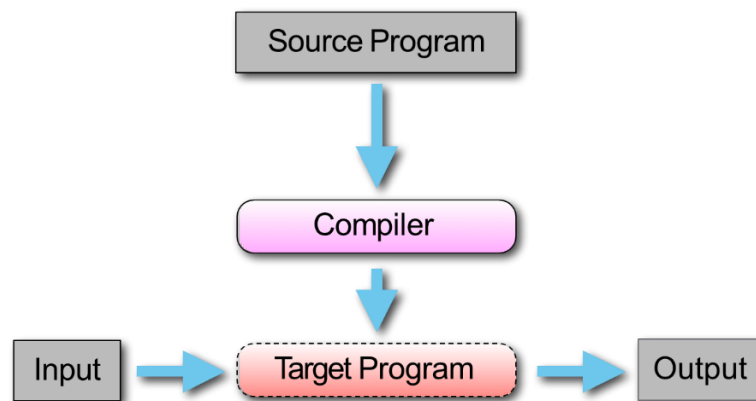
Interpretation

An interpreter is also a program that translates a high-level language into a low-level one, but it does it at the moment the program is run. You write the program using a text editor or something similar, and then instruct the interpreter to run the program. It takes the program, one line at a time, and translates each line before running it: It translates the first line and runs it, then translates the second line and runs it etc.

Interpreter characteristics:

- Relatively little time is spent analyzing and processing the program
- The resulting code is some sort of intermediate code
- The resulting code is interpreted by another program
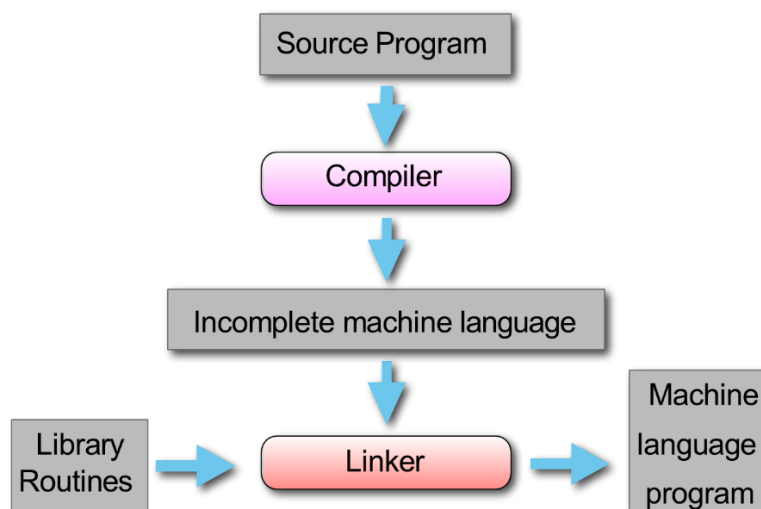- Program execution is relatively slow
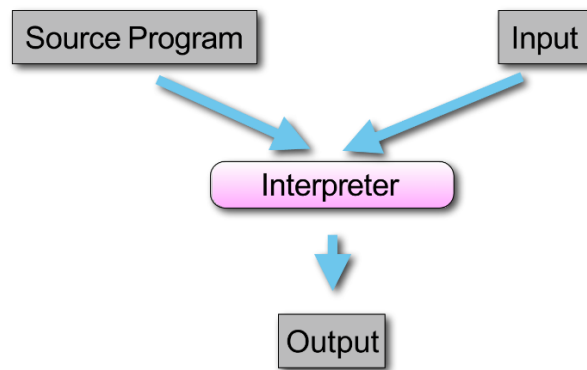
Compilation and interpretation
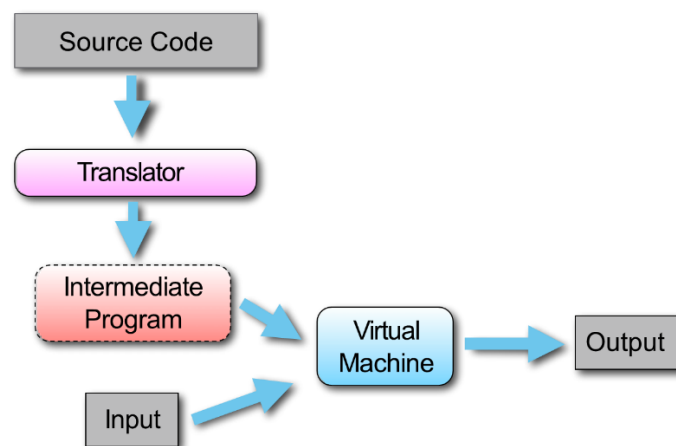
Linking

In computing, a linker or link editor is a computer system program that takes one or more object files (generated by a compiler or an assembler) and combines them into a single executable file, library file, or another 'object' file.

Preprocessing

Preprocessors are a way of making text processing with your C/C++ program before they are actually compiled. Before the actual compilation of every C/C++ program it is passed through a Preprocessor. The Preprocessor looks through the program trying to find out specific instructions called Preprocessor directives that it can understand. All Preprocessor directives begin with the # (hash) symbol. C++ compilers use the same C preprocessor.
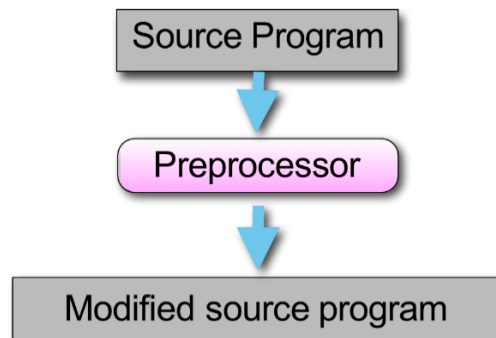
Exercises

1. Make a small report where you will describe the most known C++ compilers and interpreters available.
2. Install a C++ compiler on your personal computer:
   - On Windows 10 use preferably Visual Studio 19 Community or Enterprise edition
   - On MacOS use preferably Xcode (freely available on the apple store)
   - On Linux use preferably Code::Blocks