CMPSC 4473: Theory of Programming Languages

Compiler: A *compiler* is a computer program that takes an input source program and produces as output a sequence of machine instructions. The source program is written in a programming language such a C, C++, Java, *etc*. The machine instructions are instructions that run on a specific piece of hardware, *e.g.*, Intel x86, ARM, *etc*.

A compiler is a complex piece of software and it is often divided into the following components (or phases).

1. **Lexical analysis**: Read input, one character at a time, and group a sequence of one or more characters into an atomic unit called a *token*.
2. **Syntax analysis (parser)**: Group tokens together into syntactic structures. For example, the three tokens, *a*, + and *b*, in the input sequence *a+b* are grouped into a syntactic structure called an *expression*. Expressions might be further combined to form *statements*.
3. **Intermediate code generation**: Use the structure produced by the syntax analysis component (parser) to create a stream of simple instructions. The instructions might be similar to assembly language instructions but often do not specify registers.
4. **Code optimization**: This is an optional component that is designed to improve the intermediate code to enable the final machine instruction code to run faster and/or use less space.
5. **Code generation**: Produces the object code (machine instructions) that runs on a computer. The code generation component selects the memory and registers used in the machine instructions. Designing a code generator to produce efficient object code is one of the most difficult parts of compiler design.

Figure 1, below, depicts the components (phases) of a compiler.

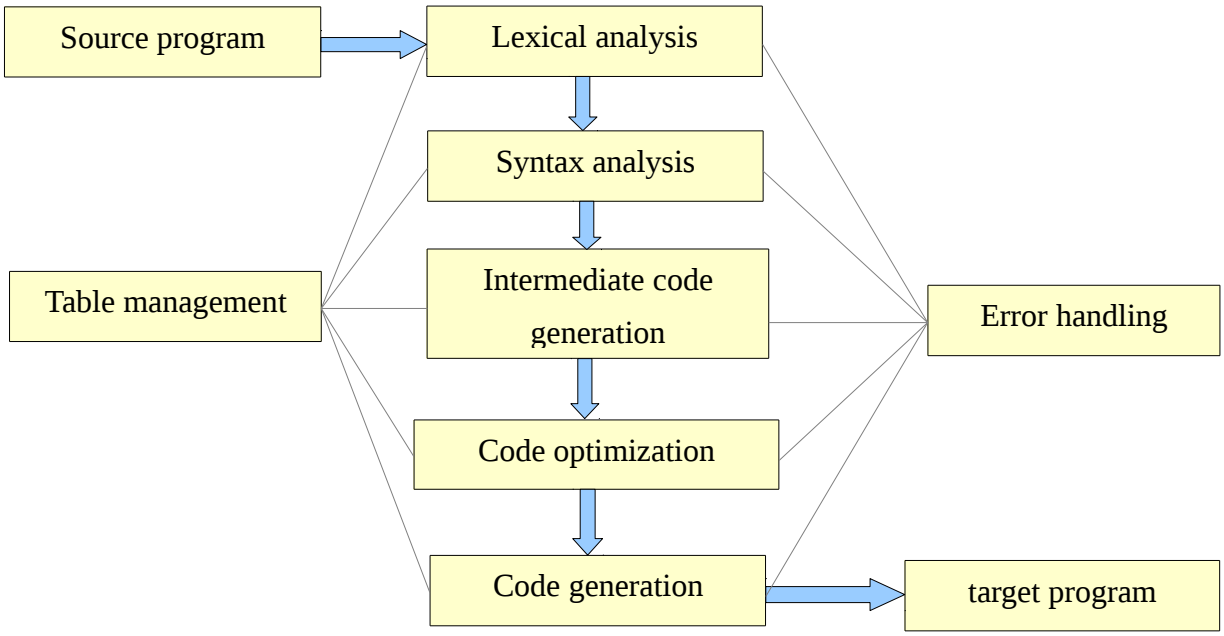

Figure 1

In figure 1 above the secondary, but important, components named *table management* and *error handling* can be associated with the other mainstream compiler components. The table management phase manages tables such as symbol tables where constants, variables, *etc*. are stored and the error handling phase detects and reports errors such as syntax errors in the source program.