Chapter 1

1.1.1 Excercises for Section 1.1

Excercise 1.1.1

Compiler: Generates a target program, in which the user can call to process inputs and produce outputs.

Interpreter: Directly executes operations on the inputs supplied from the user.

Excercise 1.1.2

- Compiler over Interpreter: Faster execution of programs
- Interpreter over Compiler: Better error reporting

Excercise 1.1.3

Assembly language is easier to produce and debug, in case the compiler has errors in it's manufacturing.

Excercise 1.1.4

Using C as a target language is helpful, since it is low level enough to manage memory directly, but high level enough to be easy to read and find errors in the compiler. Another advantage is that C has great compilers that can be used to compile it to machine code.

Excercise 1.1.5

The assembler needs

- to transate assembly to machine code,

- to resolve jumps and external memory adresses.

1.3.3 Excercises for Section 1.3

Excercise 1.3.1

1. C
    - Imperative
    - Von Neumann
    - Third-Generation
2. C++
    - Imperative
    - Von Neumann
    - Object-oriented
    - Third-Generation
3. Cobol
    - Imperative
    - Von Neumann
    - Third-Generation
4. Fortran
    - Imperative
    - Von Neumann
    - Third-Generation
5. Java
    - Imperative
    - Von Neumann
    - Object-oriented
    - Third-Generation
6. Lisp
    - Declarative
    - Von Neumann

- Functional
- Third-Generation

7. ML
    - Declarative
    - Von Neumann
    - Functional
    - Third-Generation
8. Perl
    - Imperative
    - Von Neumann
    - Object-oriented
    - Functional
    - Third-Generation
9. Python
    - Imperative
    - Von Neumann
    - Functional
    - Object-oriented
    - Third-Generation
    - Scripting
10. VB
    - Imperative
    - Von Neumann
    - Third-Generation

1.6.8: Excercises for Section 1.6

Excercise 1.6.1

```
w = 13
x = 9
```

```
y = 13
z = 9
```

Excercise 1.6.3

```
w = 9
x = 7
y = 13
z = 7
```

Excercise 1.6.3

```
w defined in B1 has scope B1 - B2
x defined in B1 has scope B1
z defined in B1 has scope B1, B3
y defined in B1 has scope B1 - B4
x defined in B2 has scope B2
z defined in B2 has scope B2 - B3
w defined in B3 has scope B3
x defined in B3 has scope B3
w defined in B4 has scope B4 - B5
x defined in B4 has scope B4 - B5
y defined in B5 has scope B5
z defined in B5 has scope B5
```

Excercise 1.6.3

```
3
2
```