Overview of Compiler Phases

Chapter 1

**Exercise 1: What are the differences between a compiler and an interpreter?**

1. Compiler: When the program's source code is debugged, it generates syntax errors.

Comparatively faster, it takes an entire program at a time.

1. Interpreter: As the user enters each line of code, syntax errors are generated.

Slower, It takes a single line of code or instruction at a time.

**Exercise 2: What are the advantages of: (a) a compiler over an interpreter (b) an interpreter over a compiler?**

1. The compiler is faster than the interpreter.

The compiler prevents the end user from viewing your source code, which is important for commercial code.

1. When compared to a compiler, an interpreter requires less memory space.

Debugging should be easier in an interpreter since it executes instructions one by one, whereas debugging in a compiler is difficult because it does not execute instructions one by one.

**Exercise 3: What advantages are there to a language-processing system in which the compiler produces assembly language rather than machine language?**

1. It is easy for debugging than machine language
2. In comparison to machine language, it is simpler to understand and apply.

**Exercise 4: A compiler that translates a high-level language into another high-level language is called a source-to-source translator. What advantages are there to using C as a target language for a compiler?**

1. The C language is easier to learn than any "proprietary" intermediate language.
2. C compilers are available for any platform, implying that your language is available on any platform and design wherever C is available.

Exercise 5: Describe some of the tasks that an assembler needs to perform.

1. - The assembler uses its source code to generate an assembly language program.
2. - allowing device drivers direct hardware access.
3. - generate machine code.

Exercise 6: Explain the terms:

1. **Translator**: A translator is anything that turns one programming language into another.
2. **Source language – object language**: The source language is the one being translated, whereas the target language, also known as the receptor language, is the one being translated into.
3. **Compiler**: In a single session, convert high-level language code to machine (object) code.
4. **Interpreter**: type of translator that turns high-level programming language code into intermediate or machine code for the machine to execute.
5. **Assembler**: program that transforms assembly language programs into machine code. It is essentially a compiler for the assembly language, but it can also be used interactively like an interpreter.
6. **Preprocessor**: directives alter the text of the source code, resulting in new source code that does not contain these directives.

Exercise 7: Describe, using an explanatory diagram, the most important parts of a compiler. The description should be based on function and input/output.

Exercise 8: What is a pass? Why are passes divided up in some cases and which factors affect this separation? Syntax analyzer

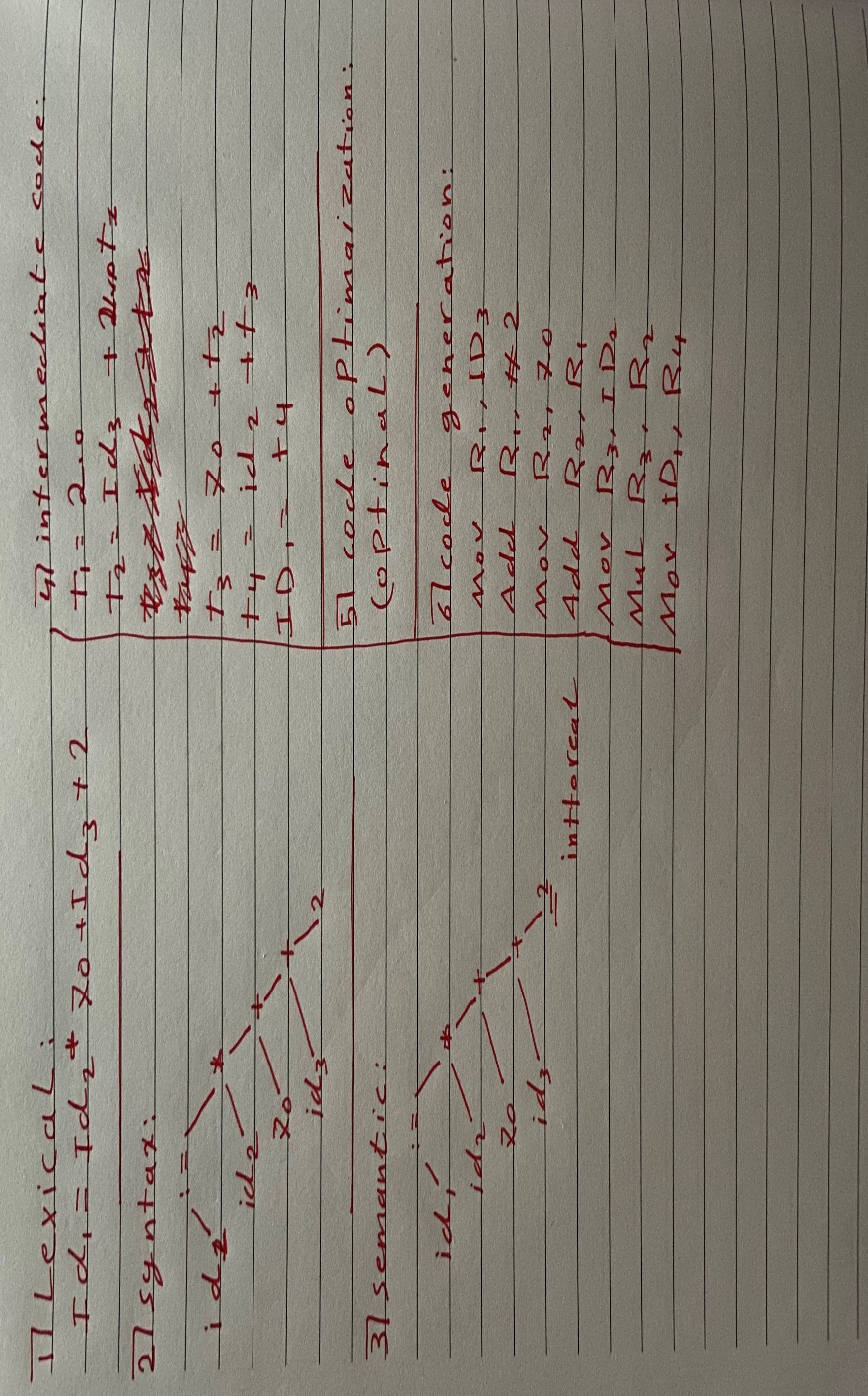
First: Each pass reads the source program and writes output into an intermediate file, which then can be read by subsequent passes, i.e., the output of one pass will be input to the next pass.

Second: It converts the program into one or more intermediate representations in steps between source code and machine code. It reprocesses the entire compilation unit in each sequential pass.

* First Pass: is refers as
  + (a). Front end
  + (b). Analytic part
  + (c). Platform independent
* Second Pass: is refers as
  + (a). Back end
  + (b). Synthesis Part
  + (c). Platform Dependent

Third: in each pass, the code improves until the final pass generates the final code. A multipass compiler performs additional tasks such as intermediate code generation, machine dependent code optimization and machine independent code optimization.

Exercise 9: Consider the following source program that contains the assignment statement



References:

1. The text book on bb
2. <https://www.codeproject.com>
3. <https://www.geeksforgeeks.org>
4. https://www.tutorialspoint.com