## Quiz 2

### 1. What are the compiler components?

From the architecture, the compiler is consists of the components below.

- Scanner ( lexical analysis)
- Parser (syntax analysis)
- Semantic Analysis
- IC generator
- Code Optimizer
- Code Generator

### 2. Briefly explain each component of the compilers?

- **Scanner**: It's a part of front end processing. It converts the original program's stream of characters into a stream of tokens. Spaces or comments in the program are removed in this process.

- **Parser**: It's a part of front end processing. It turns the token sequence into an abstract syntax tree (ATS).

- **Semantic Analysis**: It's a part of front end processing. It checks the grammar of the program is correct or not. For example, the mistake of type declaration is detected here.

- **IC generator**: It's a part of front end processing. It produces a flow graph made up of tuples grouped into basic blocks.

- **Code Optimizer**: It's a part of back end processing. It cleans up and improves the code.

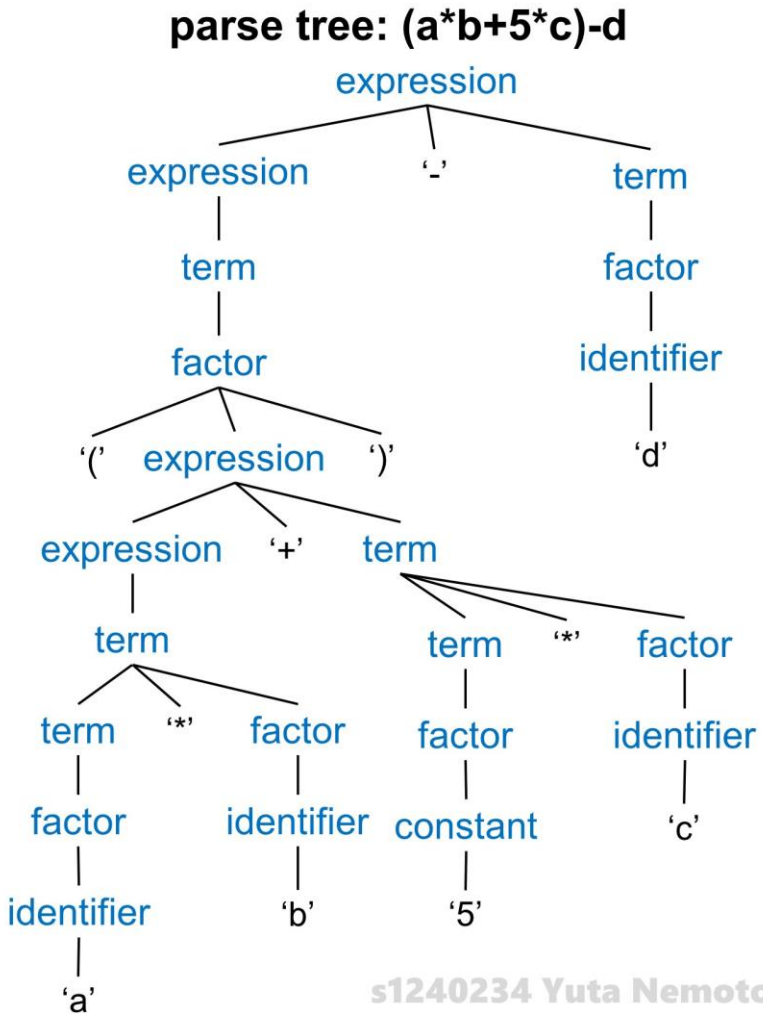
- **Code Generator**: It's a part of back end processing. It generates the actual target code. For example, the assemble code from the gcc compilation is completed here.

3. For the following grammar what is the parse tree of the expression (a*b+5*c)-d

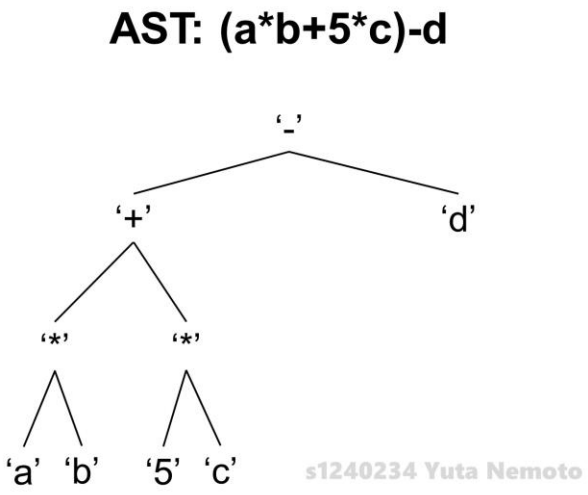
expression → expression '+' term | expression '-' term | term

term → term '*' factor | term '/' factor | factor

factor → identifier | constant | '(' expression ')'




4. Refer to the above grammar, what is the abstract syntax tree AST for the expression (a*b+5*c)-d?



5. **What is the advantage of using the compiler's front-end and back-end?**

   There are 2 principals of advantages: Retargeting and Optimization.

   1. **Retargeting**: We can build a compiler for a new machine by attaching a new code generator to an existing front-end.
   2. **Optimization**: We can reuse intermediate code optimizers in compilers for different languages and different machines.

6. **Suppose you want to write 4 compilers for 5 computer platforms, how many programs you need to write?**

   In case of direct translation, I need to write 4 * 5 = **20** programs.

   But by separating the compilation process into 2 levels: Front-end and Back-end and connecting them by Intermediate Representation, we can do this better and we need to write only 4 + 5 = **9** programs.

7. **For the regular expression ((0 | 1)0(0 | 1))* give 3 accepted strings of this regular expression?**

   1. 000
   2. 000000
   3. 100001