**COP 3402-Class Notes-01/08/19**

**What is system software?** Any software to work with and control hardware, ex: operating systems

**Most important:** Why system software?

-If you want to be a developer you may wanna know your tools
-Understanding how your tools work will make you a better programmer
-You are gonna see the results of some abstract topics like traverse trees

**Course objective:**

Building a compiler

Example:
C is a human language
Interpreting what that language does and how machine interpret human language and translate it to machine language which is Instruction sets (e.g. risk , mips, etc)

**Github**
Collaborate on source code, good development collaboration

**Recursion:**

Tree traversals are almost recursion
Any recursion can be shown as a tree

**Invariance:**

Invariance: such as pre conditions and post condition
If in your program you need a sort function you can assume it as a black box or post condition and you may say i assume that i have it and right now i don't care about

**Preprocessor**: processes the source code before compiling it
**Compiler**: is basically a translator for translating source code to assembly program
**Assembler**: can be a text base language, converts assembly language to machine code (binary code) .o file is created by assembler
**Linker**:like a modularization, glue all libraries together at the run-time
**Loader**: put the program on CPU (memory) to run

**Compiler Based:** on hardware, java
**Interpreter Based:** jdm, python, microsoft io, javascript, ruby
We simulate what machine does in interpreter

Hybrid compiler and interpreter→ translator

**Front-end compiler:**
Sequence of characters goes to lexer (lexical analyser) and it creates tokens, then syntax analyzer (parser) gets the tokens and creates tree
Semantic analysis most importantly does type checking.

**Back-end:**
Intermediate code generator, code optimizer (not in the scope of this class), code generator, target machine program

**Version control example:** git

**Demo:**

gcc: to compile a c program

Helloworld.c (a simple c program)

```
int main(){
        printf(“hello, world\n”);
        return 0
}
```

gcc helloworld.c → a.out

**Tips:**

-Start working on programming project as soon as possible
-Do programming projects individually
-Every phase of projects depends on the last.
-We will give you correct “.out” of each phase so you can still do well on your next project.