

Concepts of Programming Languages, CSCI 305, Fall 2021
Scanning – Ad-hoc and Table Driven, 54-55 & 61-69
Sept. 16

Parser code:

```
loop {  
    request a token from the scanner (return token and lexeme)  
    add to parse tree  
    add to symbol table  
}
```

Longest possible token rule – scanner identifies the largest possible string in getting a token

Three approaches to writing a lexical analyzer:

1. Using a tool: Express the acceptable patterns in a formal language such as regular expressions and input this into a software tool that automatically generates the lexical analyzer. (Unix lex tool is an example.)
2. Ad hoc: Draw a state transition diagram (DFA) that describes the patterns and write a program that implements the diagram.
3. Table driven: Draw a state transition diagram as above and construct a table-driven implementation of the state diagram.

For the first way, you just use the regular expressions. For the 2nd and 3rd way your regular expressions need to be translated to minimal DFAs.