CSE443
Compilers

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall
Syllabus

- Posted on website
- Academic Integrity
Departmental Policy on Violations of Academic Integrity (AI)

The CSE Department has a zero-tolerance policy regarding academic integrity (AI) violations.

When there is a potential violation of academic integrity in a course, the course director shall first notify the concerned students. This notification begins the review and appeals process defined in the University's Academic Integrity statement:

http://catalog.buffalo.edu/policies/course/integrity.html

Upon conclusion of the review and appeals process, if the department, school, and university have determined that the student has committed a violation, the following sanctions will be imposed upon the student:

§ 1. Documentation. The department, school, and university will record the student’s name in departmental, decanal, and university-level academic integrity violations databases. THE UNIVERSITY RECORD IS PERMANENT, AND CAN AFFECT YOUR JOB PROSPECTS (E.G. MEDICAL or LAW SCHOOL).

§ 2. Penalty Assessment. The standing policy of the Department is that all students involved in an academic integrity violation will receive an F grade for the course. The course director may recommend a lesser penalty for the first instance of academic integrity violation, and the adjudication committees that hear the appeal at the department, decanal and provost level may recommend a lesser or greater penalty.
Lexical Structure

```c
int main()
```
int main() {

character stream

int main();
Lexical Structure

character stream -> token stream

```c
int main(){
```

```c
id("int") id("main") LPAR RPAR LBRACE
```
Lexical Structure
tokens
- keywords (e.g. static, for, while, struct)
- operators (e.g. <, >, <=, =, ==, +, -, &, .)
- identifiers (e.g. foo, bar, sum, mystery)
- literals (e.g. -17, 34.52E-45, true, ‘e’, “Serenity”)
- punctuation (e.g. {, }, (, ), , ;)
meta vs object language

- **object language**: the language we are describing
- **meta language**: the language we use to describe the object language
meta vs object
language

- use quotes (meta vs 'object')
- punctuation (e.g. '{', '}', '(', ')', ',', ';')

- use font or font property (meta vs object)
- punctuation (e.g. {, }, (, ), , ;)
Formally, a **language** is a set of strings over some alphabet.

Ex. \{00, 01, 10, 11\} is the set of all strings of length 2 over the alphabet \{0, 1\}.

Ex. \{00, 11\} is the set of all even parity strings of length 2 over the alphabet \{0, 1\}.
Formally, a grammar is defined by 4 items:

1. $N$, a set of non-terminals
2. $\Sigma$, a set of terminals
3. $P$, a set of productions
4. $S$, a start symbol

$G = (N, \Sigma, P, S)$
languages & grammars

\(N\), a set of non-terminals
\(\Sigma\), a set of terminals (alphabet)
\[N \cap \Sigma = \{\}\]
\(P\), a set of productions of the form (right linear)
\(X \rightarrow a\)
\(X \rightarrow aY\)
\(X \rightarrow \epsilon\)
\(X \in N, Y \in N, a \in \Sigma, \epsilon\) denotes the empty string
\(S\), a start symbol
\(S \in N\)
Given a string $\alpha A$, where $\alpha \in \Sigma^*$ and $A \in N$, and a production $A \rightarrow \beta \in \mathcal{P}$, we write $\alpha A \Rightarrow \alpha \beta$ to indicate that $\alpha A$ derives $\alpha \beta$ in one step.

$\Rightarrow^k$ and $\Rightarrow^*$ can be used to indicate $k$ or arbitrarily many derivation steps, respectively.
languages & grammars

$L(G)$ is the set of all strings derivable from $G$ starting with the start symbol; i.e. it denotes the language of $G$. 
languages & grammars

Given a grammar $G$ the language it generates, $L(G)$, is unique.

Given a language $L$ there are many grammars $H$ such that $L(H) = L$. 
Phases of a compiler

Figure 1.6, page 5 of text