

# Compilers

CMPT 432

## – Lab 3

Goals	Turning tokens into sentences with your Parser
Notes	Now that we have tokens, we need to turn them into sentences that are (hopefully) valid in our language.
Resources	<i>Crafting a Compiler</i> <ul style="list-style-type: none"><li>• Read chapters 4.1-4.4, 5.1-5.3, and 7.1</li><li>• Do exercises 4.7 and 5.2c (write pseudo code only for 5.2c)</li></ul> <i>Dragon</i> <ul style="list-style-type: none"><li>• Read chapters 2.7, 2.8.2, 4.2, 4.4.1, and 5.3.1</li><li>• Do exercises 4.2.1 a, b, and c</li></ul>
Submitting	Commit a PDF of your work to your GitHub repository and I'll take a look at it.

---

### CHAPTER THREE. PARSING

---

1 $S \rightarrow S ; S$	4 $E \rightarrow id$	
2 $S \rightarrow id := E$	5 $E \rightarrow num$	8 $L \rightarrow E$
3 $S \rightarrow print ( L )$	6 $E \rightarrow E + E$	9 $L \rightarrow L , E$
	7 $E \rightarrow ( S , E )$	

`id := num; id := id + (id := num + num, id)`

where the source text (before lexical analysis) might have been

```
a := 7;
b := c + (d := 5 + 6, d)
```

---

### 3.1. CONTEXT-FREE GRAMMARS

---

$\underline{S}$   
 $S ; \underline{S}$   
 $\underline{S} ; id := E$   
 $id := \underline{E} ; id := E$   
 $id := num ; id := \underline{E}$   
 $id := num ; id := E + \underline{E}$   
 $id := num ; id := \underline{E} + ( S , E )$   
 $id := num ; id := id + ( \underline{S} , E )$   
 $id := num ; id := id + ( id := \underline{E} , E )$   
 $id := num ; id := id + ( id := E + E , \underline{E} )$   
 $id := num ; id := id + ( id := \underline{E} + E , id )$   
 $id := num ; id := id + ( id := num + \underline{E} , id )$   
 $id := num ; id := id + ( id := num + num , id )$

---

#### DERIVATION 3.2.

---

from *Modern Compiler Implementation in Java* by Andrew Appel