

Lecture 12 - Oct. 25

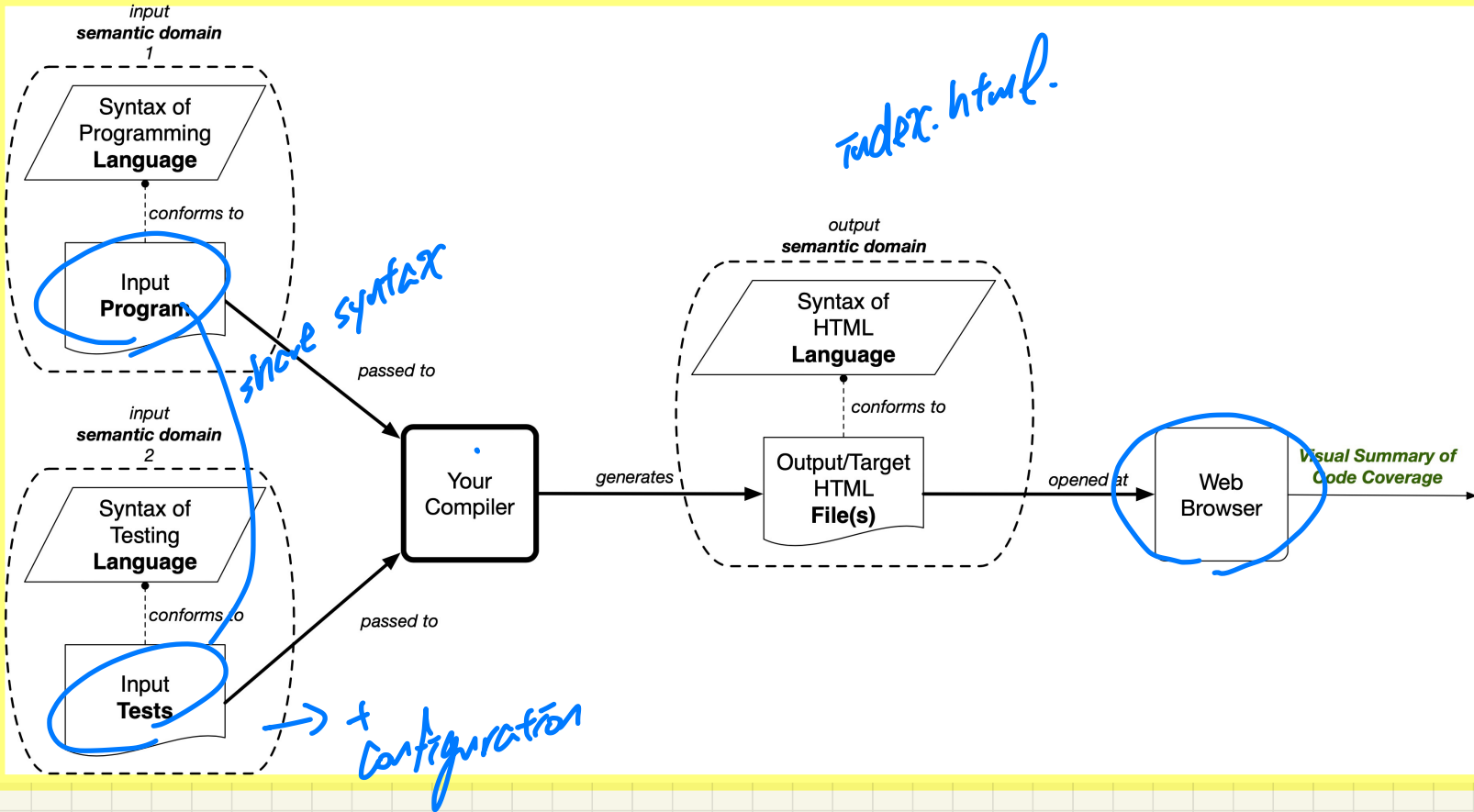
Syntactic Analysis

Derivations vs. Parse Trees
Ambiguity, Dangling else

Announcements

- **Project** teammate: Jovan
- **Programming Test**
 - + 2:00pm to 3:20pm on Saturday, October 29
 - + Venue: LAS1006 (the large lab)
- **Exam** confirmed by the registrar office:
 - + 2pm to 5pm, Saturday, December 10
 - + Last Class: Tuesday, December 6
 - + Review session?
- Updated **Calendar**
- **Quiz 3**
- **Project Specification**

Project: Roadmap



Project: Example

Example. Say you have two input files (one for program and one for tests):

```
/* Input Program */
integer absolute_value_of(integer i)
do
  if(i >= 0) then
    return i.
  else
    return -i.
  end
end
```

AST₁

```
/* Input Tests */
test_1:
  absolute_value_of(23)
```

AST₂

Interpreter/Simulator

for (int i from 10 to 20)

Then the produced output file `index.html` may be, assuming that your compiler only supports the statement/line coverage:

```
<!-- Output HTML file -->
Result of statement coverage
=====
test_1 (5/8 lines covered):
✓ integer absolute_value_of(integer i)
✓   do
✓     if(i >= 0) then
✓       return i.
     else
       return -i.
     end
✓ end

Overall Coverage: 62.5%
```



- variable assignments
(expressions)
- if-then-else
while ()
- loops
↳ bounded loop.

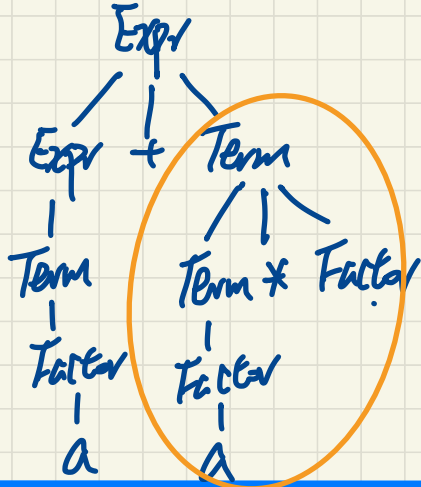
- Grading of computer ^(part of)

① run examples supplied by you

② modify/create examples based on
(a) your supplied examples
(b) supported syntax

Context-Free Grammar (CFG): **Leftmost** Derivation

Parse Tree: $a + a * a$



LMD: $a + a * a$

\Rightarrow Expr + Term
 \Rightarrow Term + Term
 \Rightarrow Factor + Term
 \Rightarrow a + Term

can be applied before +
 Term must be evaluated first
 \Rightarrow a + Term * Factor
 \Rightarrow a + Factor * Factor
 \Rightarrow a + a * Factor
 \Rightarrow a + a * a

Order of evaluation?

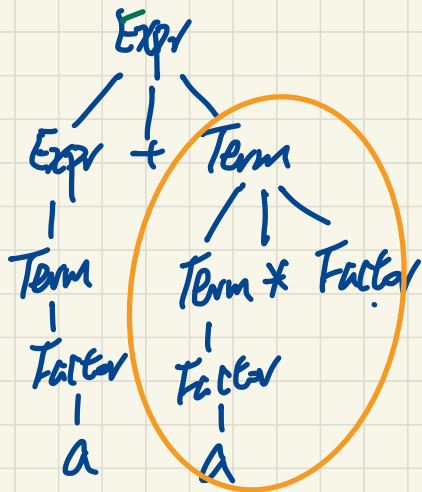
A parse tree may correspond to:

+ multiple derivations $\Rightarrow a + F * F$
 + a unique LMD $\Rightarrow a + F * a$
 $\Rightarrow a + a * a$

<u>Expr</u>	\rightarrow	<u>Expr</u> + <u>Term</u>
		<u>Term</u>
<u>Term</u>	\rightarrow	<u>Term</u> * <u>Factor</u>
		<u>Factor</u>
<u>Factor</u>	\rightarrow	(<u>Expr</u>)
		a

Context-Free Grammar (CFG): Rightmost Derivation

Parse Tree: $a + a * a$



RMD: $a + a * a$ (Exercise)

$Expr$	\rightarrow	$Expr + Term$
		$Term$
$Term$	\rightarrow	$Term * Factor$
		$Factor$
$Factor$	\rightarrow	$(Expr)$
		a

Order of evaluation?

A **parse tree** may correspond to:
+ multiple **derivations**
+ a unique **RMD**

Q. A grammar is ambiguous

if there's a string

for which there are two or more
derivations.

A. ?
=

Context-Free Grammar (CFG): Exercise (1)

Is the following CFG ambiguous?

$Expr \rightarrow Expr + Expr \mid Expr * Expr \mid (Expr) \mid a$

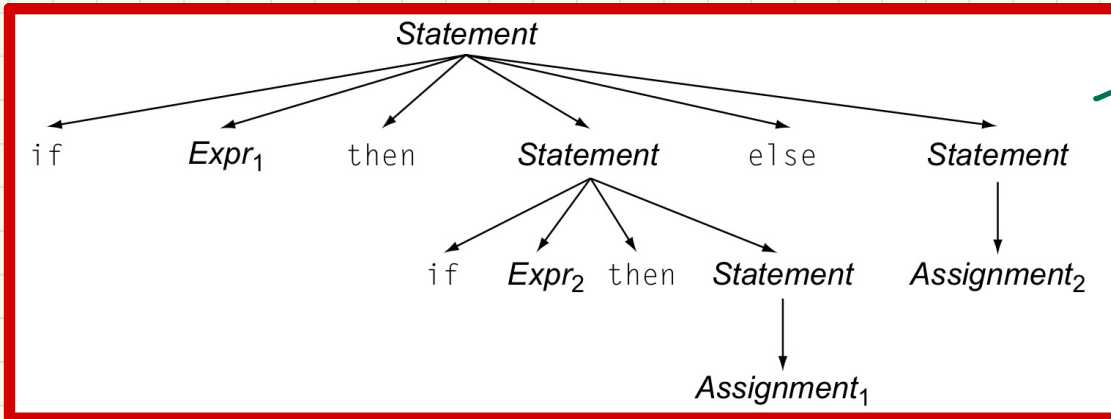
Context-Free Grammar (CFG): Exercise (2.1.1)

Is the following **CFG ambiguous**?

```
Statement → if Expr then Statement
           | if Expr then Statement else Statement
           | Assignment
           | ...
```

Example: A Possible **Semantic Interpretation?**

if Expr1 **then** **if** Expr2 **then** Assignment1 **else** Assignment2



→ Exercise:
Use two distinct
LMDs to show.

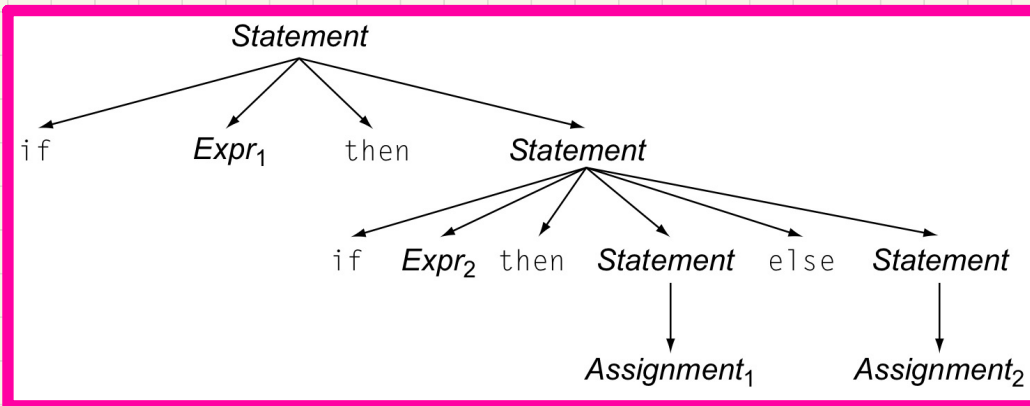
Context-Free Grammar (CFG): Exercise (2.1.2)

Is the following CFG ambiguous?

```
Statement → if Expr then Statement
           | if Expr then Statement else Statement
           | Assignment
           | ...
```

Example: A Possible **Semantic Interpretation**?

if Expr1 **then** **if** Expr2 **then** Assignment1 **else** Assignment2



Context-Free Grammar (CFG): Exercise (2.2)

Is the following CFG ambiguous?

```
Statement → if Expr then Statement  
           | if Expr then WithElse else Statement  
           | Assignment  
WithElse  → if Expr then WithElse else WithElse  
           | Assignment
```

Example: How many possible **semantic interpretations**?

if Expr1 then if Expr2 then Assignment1 else Assignment2

Can a derivation starting with **Statement** work?

Can a derivation starting with **WithElse** work?

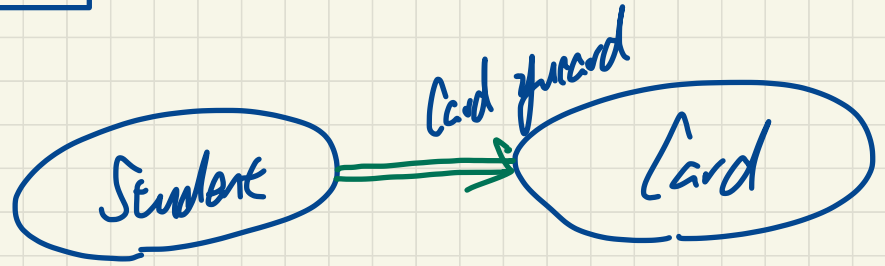
Motivation Problem: **Recursive** Systems



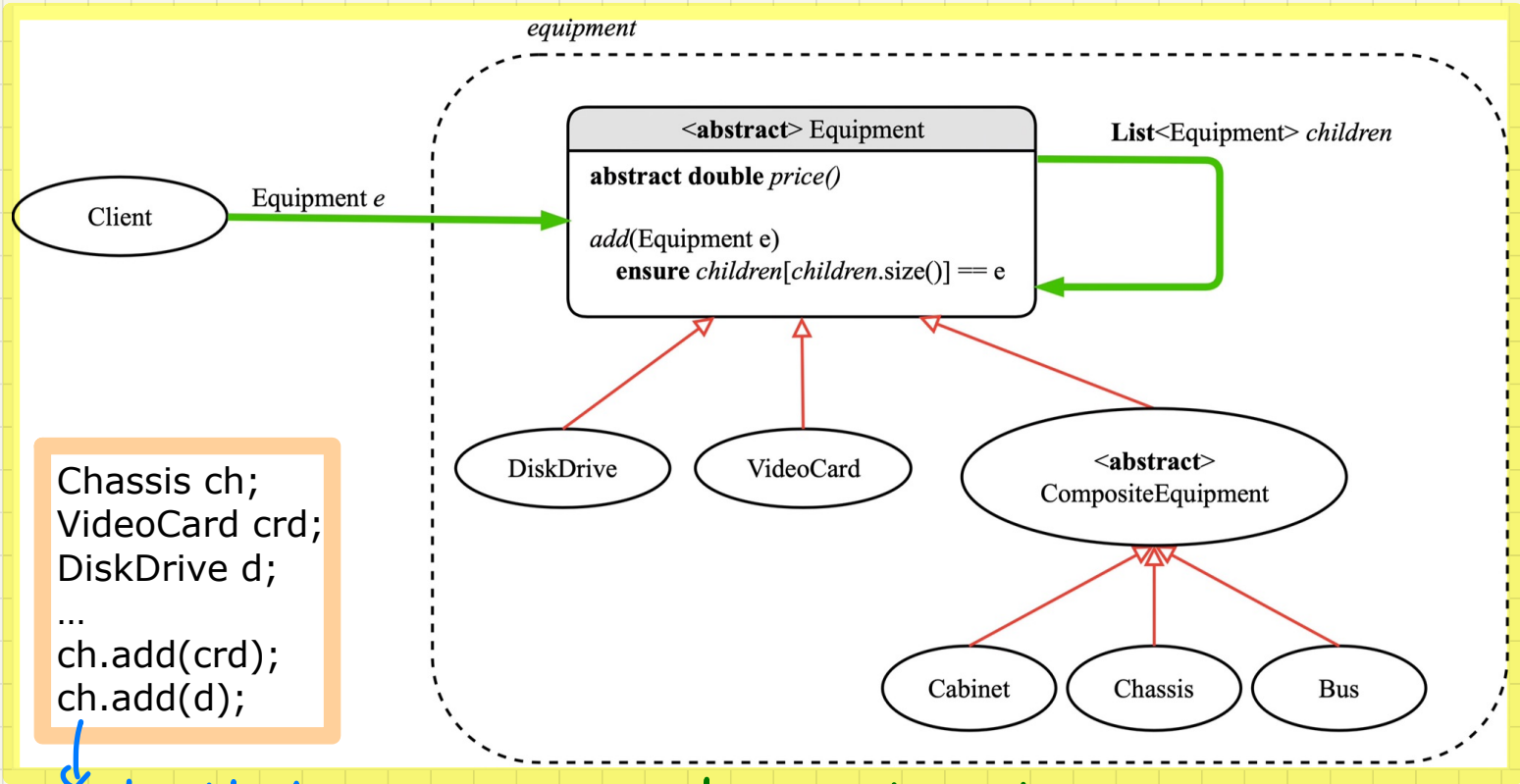
```
class Student {  
    Card yucard =  
    :  
}
```

client

supplier

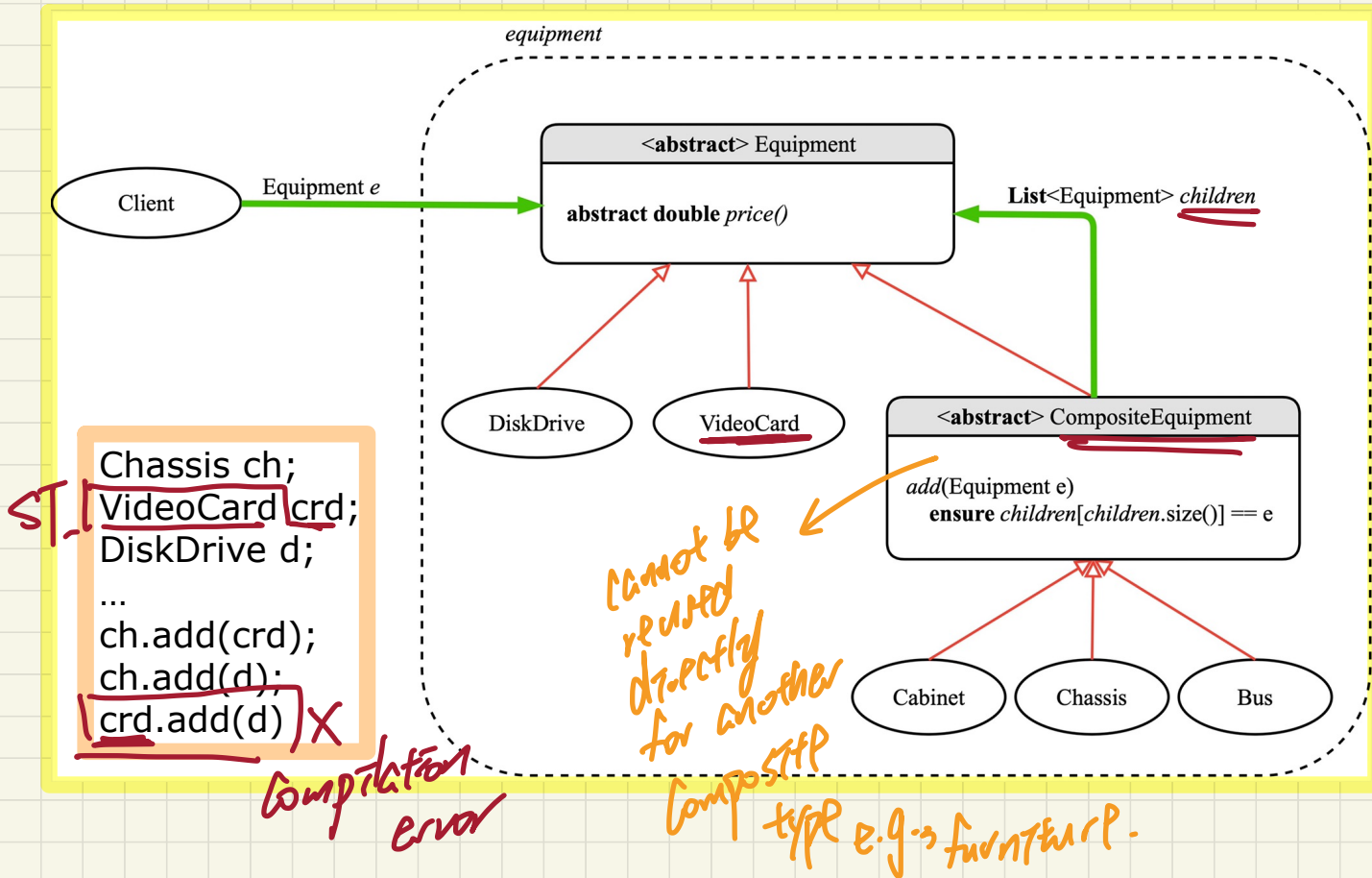


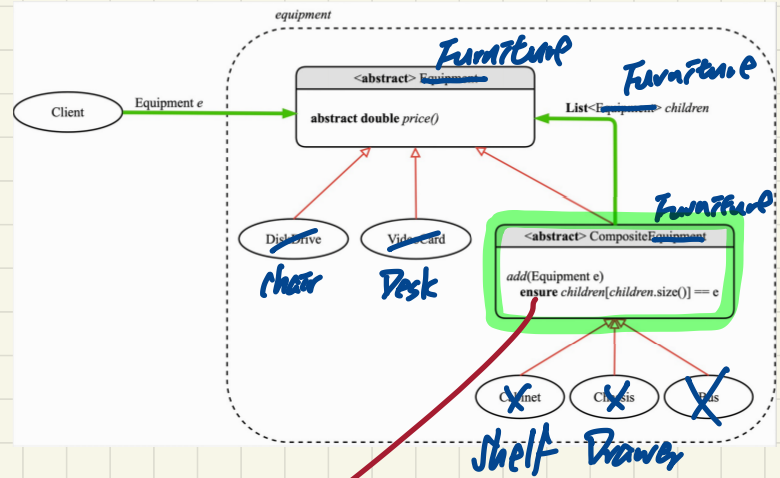
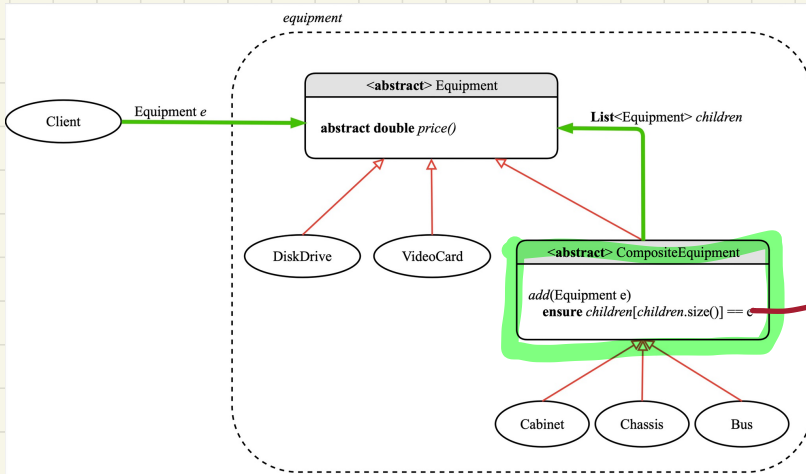
First Design Attempt



`crd.add(d)` supported but doesn't make real sense

Second Design Attempt





duplicated
 code
 => the design smells!