

# CS352 Compilers: Principle and Practice

## Spring 2005

PSO Session on Monday, January 31, 2005

By Mummoorthy M

---

**NOTE: Project 1 due this week**

**Topics:**

- a. Operator Precedence
- b. Left Recursion
- c. HashTable
- d. Perfect Programming - 2

**a) Operator Precedence:**

<http://www.cs.purdue.edu/homes/hosking/352/notes/03-parse.pdf>  
Take a look at the lecture slides. Page 3.

$x+2*y$

can be  $x+(2*y)$  - correct  
can be  $(x+2)*y$  - incorrect

"Modify the grammar that enforces precedence on the derivation"

**b) Left Recursion:**

<http://www.cs.purdue.edu/homes/hosking/352/notes/03-parse.pdf>  
Page 24.

Top-down parsing:

Grammar:  $A \rightarrow Aa$

parsing:

A  
Aa  
Aaa  
Aaaa  
Aa.....

It can go on expanding while parsing..

Let's consider two examples:

**Example 1:**

A class is defined as 'Class & a student' or just a 'student'

**Class = Class student | student**

Expansion gives us:

**(student)+**

Input:

student student student

Parsing:

Class

(Class student)

((Class student)student)

((student)student)student)

Got it!

But what's the problem? Left recursive!

Modified Grammar:

A class is a 'student and a class' or just a student.

**Class -> student Class | student**

Check if it gives (student)+ ?

Now,

student student student

Class

(Student Class)

(Student (student class))

(student (student (student)))

Got it!

## Example 2:

A train is 'Train & a bogie' or just an Engine.

**Train -> Train bogie | Engine**

Is there a left recursion? If yes, why?

What we get from this grammar?

**Engine(bogie)\***

Now, consider this:

**Train -> Engine MiniTrain**  
**MiniTrain -> Bogie MiniTrain | Nothing**

What this gives? Check if it is same as the above?

Consider a grammar

**A=Aa | b** ( same pattern as Train. Double check)

Left Recursive? yes.

It accepts: A=ba\*

input: baaa

A

(Aa)

(Aa)a

((Aa)a)a

baaa

Change the grammar:

**A->bA'**  
**A'->aA'|nothing**

Compare this with the Train example.

Can you do more? Work on the Straight Line grammar.

c. <http://java.sun.com/j2se/1.3/docs/api/java/util/Hashtable.html>

***You can use HashMap too.***

```
Hashtable numbers = new Hashtable();  
numbers.put("one", new Integer(1));
```

To retrieve a number, use the following code:

```
Integer n = (Integer)numbers.get("two");  
if (n != null)  
    System.out.println("two = " + n);
```