

CS 460

Programming Languages

Fall 2008

Dr. Watts

(Week 12)



Project 3 Output Specification



- **A listing of the original source code with lexical and syntactical error messages after the line on which the error appears.**
- **If there are no errors, list name, type and value for each identifier in the source program after the complete program.**
- A debugging file (filename + .dbg) containing a list of terminal and non-terminal symbols encountered while parsing the program (and other useful debugging information).



Error Messages

- ERROR: Invalid token.
- Error Found: Missing First in <stmt_tail>
- unexpected '='
[4] a *= b; a /= b; a %= b; a != b; a && b; a &= b;
- Error near this position 21:35
- ERROR: line: 12 position 15:
Expecting SEMI token RPAREN
- Unexpected '|=' found near 5:12



Error Message Haiku

Errors have occurred.
We won't tell you where or why.
Lazy programmers.

-- Charlie Gibbs

<http://archive.salon.com/21st/chal/1998/02/10chal2.html>

Symbol and Tables



- What is in a symbol?
- Table
 - Collection of symbols

Stack-based evaluation



- Example 1:
 - $A = 6 + 3 * 4 - 5;$
- Example 2:
 - $B = 5;$
 - $C = 7;$
 - $D = A + B - 3.2 * C;$
- Example 3:
 - $D += 3 * B - 4 * A;$



Stack-based evaluation

- Example 4:
 - $A = 6 + 3 * (4 - 5 / 2) + 3 * 7;$
- Example 5:
 - $B = (6 + 3) * ((4 - 5 / 2) + 3 * 7);$
- Example 6:
 - $C = 7;$
 - $D = (A + B) - (3.2 * C - (2 + A));$
- Example 7:
 - $D += 3 * (B - 4) * (A - (4 + C) * 2);$



Stack-based evaluation

- Example 8 input:
 - $A = 5;$
 - $B = 7;$
 - $C = 3 * (A + 2) * B;$
 - $D = 7;$
 - $D += 4 * (B + 2.3) * (C + A);$



Stack-based evaluation

- Example 8 output:

```
[1] A = 5;  
[2] B = 7;  
[3] C = 3 * (A + 2) * B;  
[4] D = 7;  
[5] D += 4 * (B + 2.3) * (C + A);  
0 errors found.  
A      5  
B      7  
C     147  
D    5661.39990
```



Stack-based evaluation

- Example 8 C version:

```
#include <stdio.h>  
  
int main ()  
{  
    int A, B, C;  
    float D;  
    A = 5;  
    B = 7;  
    C = 3 * (A + 2) * B;  
    D = 7;  
    D += 4 * (B + 2.3) * (C + A);  
    printf ("A\t%d\n", A);  
    printf ("B\t%d\n", B);  
    printf ("C\t%d\n", C);  
    printf ("D\t%7.5f\n", D);  
}
```



Annotated Grammar

- Symbol Creation
 17. `<var>` -> IDENT [create-symbol]
 18. `<var>` -> NUMLIT [create-symbol]
- Symbol Output
 1. `<program>` -> `<stmt>` SEMI `<more_stmts>` EOFT [print-symbols]
- Statement Queue Manipulation
 1. `<program>` -> `<stmt>` [evaluate-queue] SEMI `<more_stmts>` EOFT
 2. `<more_stmts>` -> `<stmt>` [evaluate-queue] SEMI `<more_stmts>`
 6. `<term>` -> LPAREN `<stmt>` [evaluate-queue] RPAREN
 9. `<uoppre>` -> PLUS [add-symbol-to-queue]
 10. `<uoppre>` -> MINUS [add-symbol-to-queue]
 17. `<var>` -> IDENT [create-symbol] [add-symbol-to-queue]
 18. `<var>` -> NUMLIT [create-symbol] [add-symbol-to-queue]
 19. `<post>` -> PLUSPLUS [add-symbol-to-queue]
 20. `<post>` -> MINUSMINUS [add-symbol-to-queue]
 25. `<binop>` -> PLUS [add-symbol-to-queue]
 26. `<binop>` -> MINUS [add-symbol-to-queue]