

CSE 305 Programming Languages Spring, 2005

Homework 2

Maximum Points: 24

Due 10:00 am, Monday, February 7, 2005 at the start of Lecture

Professor Shapiro

January 31, 2005

Write the answers to this homework set on one or more 8.5×11 inch sheets of paper, and turn them in at the start of lecture on the date given above. If you need more than one sheet of paper, use one side only, and staple the sheets together at the upper left-hand corner. Identify your answers clearly with the course, and your name. Illegible answers will be considered incorrect.

If your answer to some question is based on some source material, including a book or web site, cite the source as part of your answer.

The point value of each question is shown in parentheses.

1. (3) (3.2a of *Sebesta*) Write an EBNF grammar for a Java class definition header. For example, the class definition header of the program `Echo.java`, from hw1, is

```
public class Echo
```

However, note that Java headers can be more complicated than this.

You should answer this question by referring to §8.1 of *The Java Language Specification, Second Edition*: http://java.sun.com/docs/books/jls/second_edition/html/classes.doc.html#15372, and translating their BNF style to ours. Write an EBNF grammar whose topmost nonterminal symbol is $\langle \text{class definition header} \rangle$, and whose leaf symbols are either terminal symbols, the nonterminal symbol $\langle \text{Identifier} \rangle$, or the nonterminal symbol $\langle \text{TypeName} \rangle$. Do not include in your grammar the grammar of the class definition body—just the header.

2. (3) (3.3 of *Sebesta*) Rewrite the BNF of Example 3.4 (p. 123) to give + precedence over * and force + to be right associative.
3. (3) (Based on 3.7 of *Sebesta*) Using the original, unmodified grammar in Example 3.4 (p. 123), show a parse tree for the statement,

```
A = B * (C * A + B)
```

4. (3) Using the following grammar (based on *Sebesta*, p. 128):

```
<stmt> -> <matched> | <unmatched>
<matched> -> if <logic_expr> then <matched> else <matched>
           | <non-if-stmt>
<unmatched> -> if <logic_expr> then <stmt>
           | if <logic_expr> then <matched> else <unmatched>
```

show a parse tree for this sentential form (based on *Sebesta*, Figure 3.5, p. 127):

```
if <logic_expr> then if <logic_expr> then <non-if-stmt> else <non-if-stmt>
```

5. (3) Write any $\langle \text{if statement} \rangle$ that is legal according to the following grammar:

```
 $\langle \text{statement} \rangle$            $\rightarrow$   $\langle \text{if statement} \rangle$  |  $\langle \text{assignment statement} \rangle$   
 $\langle \text{if statement} \rangle$         $\rightarrow$  if  $\langle \text{expr} \rangle$  then  $\langle \text{statement} \rangle$  else  $\langle \text{statement} \rangle$  endif ;  
 $\langle \text{assignment statement} \rangle$   $\rightarrow$   $\langle \text{id} \rangle$  :=  $\langle \text{expr} \rangle$  ;  
 $\langle \text{expr} \rangle$                  $\rightarrow$   $\langle \text{id} \rangle$   
 $\langle \text{id} \rangle$                   $\rightarrow$  i | j | k | x | y | z
```

6. (3) What is printed by the following HOSL program?

```
    i := 1  
loop: print i  
    if i = 5 goto out  
    i := i + 1  
    goto loop  
out:  print i
```

7. (3) What is printed by the following HOSL program?

```
    i := 1  
loop: if not (i <= 5) goto out  
    print i  
    i := i + 1  
    goto loop  
out:  print i
```

8. (3) Using the techniques of the class web notes

<http://www.cse.buffalo.edu/~shapiro/Courses/CSE305/2005/notes4.html>

trace the execution of the following LOSL program:

```
x y 3 store 1 + store pop stop
```