CMSI 488

Midterm

The test is open-everything with the sole limitation that you neither solicit nor give help while the exam is in progress.

Problem	You got	Out of
1		15
2		15
3		15
4		15
5		15
6		10
7		15
TOTAL		100

1. In the notes on syntax, we looked at a little language with assignments, read and write statements, while statements, and some basic arithmetic. We gave a concrete and abstract syntax for this language, too. We also saw a number of alternative styles for writing programs with the same abstract syntax but different concrete syntaxes. Give an EBNF for the **macrosyntax** of the variant of the language in the notes illustrated by the example beginning with "COMMENT THIS LOOKS LIKE OLD CODE".

2. For each of the following:

```
x x = x.x;   // a declaration


x = x.x;     // an assignment statement
```

state whether each can ever legally appear in a Java program. If it can, state what the construct means, semantically. If not, state, very precisely and technically *why* it cannot, hinting at either (1) why allowing it would be stupid (inconsistent with the language design), or (2) the inexplicable oversight or stupidity of the language designer for leaving the ability out.

3. For each of the following Java **fragments**, tell whether it would result in a lexical error, syntax error, static semantic error, dynamic semantic error, or no error — **and why**.

a.

```
class Hero {
  String name;
  java.util.Date birthday;
  boolean super;
}
```

b.

```
void f(int x) {int x = 8;}
```

c.

```
String `ohana = "family";
```

d.

```
String pet = "dog";
pet = "rat";
```

e.

```
String pet = "cat"
pet[0] = 'r';
```

4. Sketch the AST for the following Java fragment:

```
static protected synchronized long g(Object... m) {
    for (int y : f(x)) {
        x = p.data[0] * (3<<   7|-  x---c);
    }
}
```

5. Suppose we are designing a fully statically typed, strongly typed, language. We decide to have type declarations, such as

   ```
   type List is array of Integer;
   type Point is class X: Integer, Y: Integer end;
   ```

   and variable declarations such as:

   ```
   var A: List;
   var B: Point;
   var C: Integer;
   ```

   Now we want to allow array and object expressions, such as

   ```
   [6.5, 7.0, 4, "Orange"]
   {Y: 5, X: 3}
   ```

   Explain how our compiler would keep track of the types of expressions and how it would typecheck such expressions during semantic analysis.

6. Given the following grammar:

```
G -> (S s G)?
S -> V q E | r i | w E | u E S+
V -> i | V d i | V a E a
E -> n | V | a E* a | b (I E)* b
```

a. Is it ambiguous?
b. Is it LL(k)?
c. Give a JavaCC function (or functions) that can syntax check the V rule.
d. What does it represent?

7. Write regular expressions (you may use all the Java pattern features) for
    a. Octal constants in C

    b. Floating point constants that are allowed to have an empty fractional part and can have no more than three digits in the exponent part

    c. the set of all character strings that contain neither the substring "exit" nor "exec"

    d. Unsigned 32-bit binary numerals divisible by 8

    e. Non-empty words consisting of the letters a-z whose first and second halves are the same (i.e., in set notation: $\{ww \mid w \in \{\text{a..z}\}+\}$)