Ryerson University
Department of Electrical & Computer Engineering

COE808

Quiz **Jan. 28, 2015**

Name:_____ **Student #: _____**

Time: 1 hr 30 min.

Q1. For the following grammar (X, Y, and Z are nonterminals with X being the start): (6 marks)

<X> → <Y><Y><Z>
<Y> → <X>a | a
<Z> → b<Z> | b

Circle the strings that can be generated by the grammar:

(aabb) aaaab (aabaaabab)

Q2. Which string is not accepted in this grammar? (4 marks)

<real> → [+|-]<integer> . <fraction>
<integer> → <digit>| <digit> <integer> | ε
<fraction> →<digit>|<fraction><digit> | ε
<digit> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

(a) +12.
(b) 2.3
(c) 15.
(d) .23
(e) None of the above.

Q3. What will happen if we try to compile and run the following Java code ? (4 marks)

```
public class Test {
   private static int j = 20;

   public static void main(String argv[]) {
      int i = 10;
      Test t = new Test();
      t.test(i);
      System.out.println( i );
      System.out.println( j );
   }

   public void test(int x) {
      x = x*2;
      j = j*2;
   }
}
```

1) Compilation error
2) Output: 20
   40
3) Output: 10
   40
4) Output: 10
   20

Answer: 3)

Q4. Given the following grammar: (10 marks)

<assign> → <id> := <expr>
<id> → A | B | C
<expr> → <expr> + <expr>
| <expr> * <expr>
| (<expr>)
| <id>

show a parse tree and a leftmost derivation for the following statement.

A := (B+C) * C

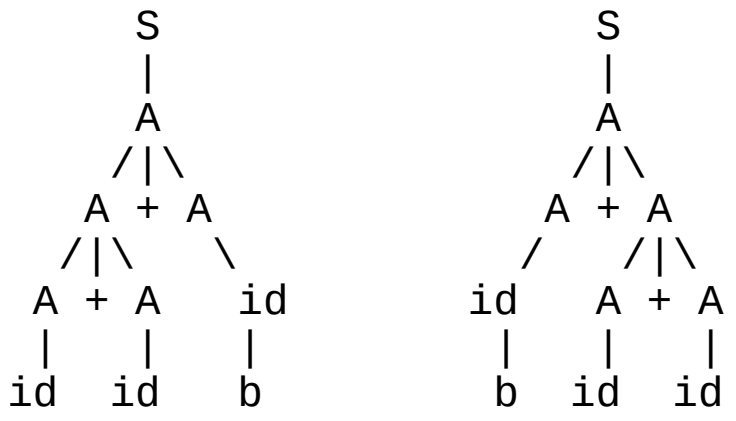
Show parse tree (5 marks)

Leftmost derivation(5 marks)

Q5. Show that the following grammar is **ambiguous** if the "b+b+b" is parsed. (10 marks)

<S> →<A>
<A> → <A> + <A> | <id>
<id> → a | b | c

Answer:
"b + b + b" has multiple parse trees:



therefore the grammar is ambiguous.

Q6. Given the following program written in a C-like syntax but which is ***not*** a C program, what variables (g, x, v-main, v-sub1, z-sub1, z-sub2, w) are visible inside of the call to foo if this language has: (6 marks)

a) static scoping?
w, z-sub2
b) dynamic scoping?
x, z-sub2, w, v-sub1

```
int main() {
int x;
int v; /* refer to this variable as v-main */
sub1();
}

void sub1 () {
int v; /* refer to this variable as v-sub1 */
int z; /* refer to this variable as z-sub1 */
sub2(g);
}

void sub2 (int z) {/* refer to this variable as z-
                      sub2 */
int w;
int foo() {
...          /* what variables are visible here? */
}
foo();
}
```

Q7. Show a complete parse, including the parse stack contents, input string, and action for the string "1+1" (10 marks)

	action						*goto*	
state	*	+	**0**	**1**	**$**		**E**	**B**
0			s1	s2			3	4
1	r4	r4	r4	r4	r4			
2	r5	r5	r5	r5	r5			
3	s5	s6			acc			
4	r3	r3	r3	r3	r3			
5			s1	s2				7
6			s1	s2				8
7	r1	r1	r1	r1	r1			
8	r2	r2	r2	r2	r2			

The grammar is as below:
(1) <E> → <E> * <B>
(2) <E> → <E> + <B>
(3) <E> → <B>
(4) <B> → 0

Answer:

Stack Input Action/Goto

Stack Input Action

0 1+1$ Shift 2
0‘1’2 + 1 $ Reduce 5 (Use GOTO[0, B])
0‘B’4 + 1 $ Reduce 3 (Use GOTO[0, E])
0‘E’3 + 1 $ Shift 6
0‘E’3‘+’6 1 $ Shift 2
0‘E’3‘+’6‘1’2 $ Reduce 5 (Use GOTO[6, B])
0‘E’3‘+’6‘B’8 $ Reduce 2 (Use GOTO[0, E])
0E3 $ Accept