CSE 307 – Assignment # 5

Assignment problem # 1 (marks: UG: 8, PG: 6)

Define a no short-circuit and short-circuit version of the following statements where && and || are the conditional operators in Java (default short-circuit operators), while & and | are the unconditional operators in Java. The & operator works the same as the && operator, and the | operator works the same as the || operator with one exception: the & and | operators always evaluate both operands, while && and || shortcut when possible.

JZ	op1	jump if zero
JNZ	op1	jump if not zero
JE	op1 = op2	jump if equal
JNE	op1 != op2	jump if not equal
JG	op1 > op2	jump if greater than
JNG	!(op1 > op2)	jump if not greater than
JGE	op1 >= op2	jump if greater than or equal
JNGE	!(op1 >= op2)	jump if not greater than or equal
JL	op1 < op2	jump if less than
JNL	!(op1 < op2)	jump if not less than
JLE	op1 <= op2	jump if less than or equal
JNLE	!(op1 <= op2)	jump if not less than or equal

A. if ((A <= B | C > D) & (E > F | G < H): I else J (short-circuit version not applicable)
B. if ((A <= B || C > D) && (E > F || G < H)): I (show both versions – no short-circuit and short-circuit)

Assignment problem # 2 (marks: UG: 8, PG: 6)

Consider the following program:

```
var total = 0
def a():
  total += 1
def b(f):
  var total = 0
  f()
  print "B", total
def c():
  var total = 0
  b(a)
  print "C", total
c()
print "T", total
```

Determine the three lines of output for the following scoping cases:

A. static scoping:
B. dynamic shallow scoping:

Assignment problem # 3 (marks: UG: 3, PG: 3)

In Lisp, most of the arithmetic operators are defined to take two or more arguments, rather than strictly two. Thus (* 2 3 4 5) evaluates to 120, and (- 16 9 4) evaluates to 3. Show that parentheses are necessary to disambiguate arithmetic expressions in Lisp (in other words, give an example of an expression whose meaning is unclear when parentheses are removed).

Assignment problem # 4 (marks: UG: 6, PG: 6)

Refer to description of infix, prefix and postfix notations at the following page:

http://www.cs.man.ac.uk/~pjj/cs212/fix.html

http://interactivepython.org/runestone/static/pythonds/BasicDS/InfixPrefixandPostfixExpressions.html

Convert the following infix expressions to prefix and postfix form:

A. 1 circle 2 star 3
B. 4 circle 5 star 6 circle 7

where the precedence and associativity chart is:

- star highest precedence and left-associative
- circle lowest precedence and left-associative

Translate the following expression into prefix and postfix notation:

C. [−b + sqrt(b × b − 4 × a × c)]/(2 × a)

Assignment problem # 5 (Only for graduate students) (marks: PG: 4)

Consider the following pseudocode:

```
x : integer := 1
y : integer := 1
procedure add
x := x + y
print x
print y
procedure second(P : procedure)
x : integer := 2
P()
print x
print y
procedure first
y : integer := 3
second(add)
print x
print y
first()
print x
print y
```

Determine the printout of this program using:

A. static scoping:
B. dynamic scoping with shallow binding: