Nalongsone Danddank Student ID : 14958950 StarID: jf3893pd

Email: [nalongsone.danddank@my.metrostate.edu](mailto:nalongsone.danddank@my.metrostate.edu)\

**Metropolitan State University**

**ICS-365 Exam #2**

**75 Points**

**PART 1. (20 points, 1 point each, Pick 20, Skip the 3 you don’t wish graded)**

1. What is the correct definition of coercion in the context of programming languages?
   1. Explicit conversion of a value into another data type.
   2. Coercion is equivalent to unary assignment operators.
   3. Implicit conversion of a value into another data type.
   4. None of the above
2. What is a conditional expression?
   1. An expression that only exists in certain conditions such as linking or building.
   2. There is no such thing as a conditional expression.
   3. A way of selectively altering the control flow based on some condition
   4. The definition of a prefix operator.
3. What is short-circuit evaluation?
   1. The name of the optimization when the compiler replaces a loop that executes exactly 5 times with 5 consecutive copies of the loop body.
   2. One that doesn't necessarily evaluate all of its operands
   3. The name given to an interpreter of byte codes
   4. The name of a node in a control flow graph.
4. Which of the following is not a real design issue for character string types.
   1. Should strings be simply a special kind of character array or primitive type?
   2. Should strings have static or dynamic length?
   3. Should strings exclude certain words or phrases?
   4. Should string operations be built-in to the language or supported by user developed libraries.
   5. All the above are real design issues.
5. Circle the 3 parts of an IEEE floating point type
   1. Sign
   2. Exponent
   3. Mantissa
   4. Precision
6. What is an enumeration type?
   1. One in which the values consist of sequences of characters.
   2. One in which all the possible values which are name are provided in the definition.
   3. One in which values are represented as ordered pairs of floating point numbers
   4. One in which data types are not defined in terms of other types.
7. What is a primitive type?
   1. One in which data types are not defined in terms of other types.
   2. One in which the values consist of sequences of characters.
   3. One in which all the possible values which are named are provided in the definition.
   4. One in which values are represented as ordered pairs of floating point numbers
8. What is a lists
   1. A type whose variables may store different data type values at different times.
   2. A data type in Python that can store different data types and is immutable
   3. A homogeneous aggregate of data elements in which an individual element is identified by its position in the aggregate.
   4. An unordered collection of data elements that are indexed by an equal number of values called keys.
9. What is a union
   1. A type whose variables may store different type values at different times.
   2. A data type that is similar to a record, except the elements are not named.
   3. An aggregate of data elements in which the individual elements are identified by names and accessed through offsets.
   4. An unordered collection of data elements that are indexed by an equal number of values called keys.
10. In scheme, nested lists are not allowed?
    1. True
    2. False
11. In scheme, the function that returns a list minus its first element?
    1. CAR
    2. CBR
    3. None of these
    4. CDR
12. In scheme, the function that returns the first element of a list?
    1. CDR
    2. CBR
    3. CAR
    4. None of the above
13. Which of the following allows has variable scope to be determined prior to execution?
    1. Dynamic scoping
    2. Static scoping
14. Which of the following is based on a calling sequence and scope can only be determined at runtime.
    1. Dynamic scoping
    2. Static scoping
15. What is a referencing environment?
    1. The collection of all variables and functions visible at a given moment of program execution.
    2. The collection of source code, object code, and libraries linked to produce a program.
    3. A variable that is bound to a value only once.
    4. The type of system a program is running on.
16. What is the definition of a control structure?
    1. Put simply, a block of code that dictates the flow of control
    2. A structure, such as a struct in c, a pic in cobol, or record in algol
    3. The general arrangement of a program.
    4. None of the above.
17. Which of the following functions implements a multiple selection statement in scheme?
    1. Quote
    2. CASE
    3. Both A and B
    4. COND
    5. None of the above
18. Which language does not have an unconditional branch?
    1. ‘c’
    2. Fortran
    3. Java
    4. All have some type of unconditional branch
19. A variable binding is static if
    1. Occurs before run time and remains unchanged throughout execution
    2. Occurs during execution, or Can change during execution
    3. Gives you a shock if you touch it.
    4. None of the above.
20. A variable binding is dynamic if
    1. Occurs before run time and remains unchanged throughout execution
    2. Gives you a shock if you look at it.
    3. Occurs during execution, or Can change during execution
    4. None of the above.
21. Given the code below, written in a fictitious programming language, what is the output.
    1. 2
    2. 4
    3. 6
    4. \_\_\_\_\_ (other value not shown)

int x = 20;

int y = 5;

switch (x % 7)

{

case 0:

case 1: y++;

case 2:

case 3: y = y + 2;

case 4: break;

case 5:

case 6: y = y – 3;

}

print y

1. What is returned by the following scheme expression: (\* 4 ( + 4 4))
   1. 32
   2. 20
   3. \_\_\_\_ other
   4. syntax error
2. What is the output of the following

(case (\* 2 3)

((2 3 5 7) 'prime)

((1 4 6 8 9) 'composite))

1. prime
2. composite
3. prime composite
4. nothing, no output

**Section 2, Problems 15 points, Do all.**

**Problem #1. (5 points each (A, B and C))**

Given the following pseudo code in a hypothetical programming environment, what are the outputs under the following conditions?

begin

integer m, n;

procedure hardy;

begin

print("in hardy -- n = ", n);

end;

procedure laurel(n\_: integer);

begin

integer n = n\_;

print("in laurel -- m = ", m);

print("in laurel -- n = ", n);

hardy;

end;

m := 50;

n := 100;

print("in main program -- n = ", n);

laurel(1);

hardy;

end;

(5 points) A. Called using static scope.

Answer:

in main program -- n = 100

in laurel -- m = 50

in laurel -- n = 1

in hardy -- n = 100

in hardy -- n = 100

(5 points) B. Called using dynamic scope

Answer:

in main program -- n = 100

in laurel -- m = 50

in laurel -- n = 1

in hardy -- n = 1

in hardy -- n = 100

**Problem 1 (continued)** (5pts for letter C)

C. Given the following pseudo code in a hypothetical programming environment, what are the outputs using static scoping only.

procedure laurel(n: integer);

begin

procedure hardy;

begin

print("in hardy -- n = ", n);

end;

print("in laurel -- m = ", m);

print("in laurel -- n = ", n);

hardy;

end;

m := 50;

n := 100;

print("in main program -- n = ", n);

laurel(1);

/\* can't call hardy from the main program any more \*/

end;

Answer:

in main program -- n = 100

in laurel -- m = 50

in laurel -- n = 1

in hardy -- n = 1

**PROBLEM 2. (10 Points, 2pts each)**

**Assume the following rules of associativity and precedence for expressions:**

**Precedence:** Highest Parentheses first

\*, /, not

+, –, mod

- (unary operation)

=, /=, <, <=, >=, >

and

Lowest or, xor, land (logical ‘and’)

**Associativity:** Left to right

Show the order of evaluation of the following expressions by parenthesizing all subexpressions and placing a superscript on the right parenthesis to indicate order. For example, for the expression **a + b \* c + d**, the order of evaluation would be represented as **((a + (b \* c)1 )2 + d)3**

1. a + b \* 2 xor c

((a + (b \* 2)1)2 xor c)3

1. a > b xor c or d <=17

(((a > b)1 xor c)3 or (d <= 17)2)4

1. a \* b –c \* d and e + f

(((a \* b)1 – (c \* d)2)3 and (e + f)4)5

1. -a + b

(-(a + b)1)2

1. –a or c + d

((–a)2 or (c + d)1)3

**PROBLEM 3. 10 Points.**

**Translate the following call to Scheme’s COND to java. Set the resulting value to ‘y’ (etc. y = ……….). Make sure to use good Java syntax.**

(COND

((> x 10) x)

((< x 5 ) (+ 2 x))

((= x 8) (\* x 10))

)

Solution:

if (x > 10) {

y = x;

} else if (x < 5) {

y = 2 + x;

} else if( x == 8) {

y = x \* 10;

}

**Problem 4. 10 Points**

**Write a scheme function findlocate which takes two parameters, an atom atm and a list, and returns the location of the first location where atm occurs in the list.**

**If atm does not occur in the list, the function returns n + 1, where n is the length of the list.**

Example 1: (findlocate 'f '(a f c g f b))

This should return 2

Example 2: (findlocate 'z '(a f c g f b))

This should return 7

Solution:

(define (atom? x)

(and (not (null? x))

(not (pair? x))))

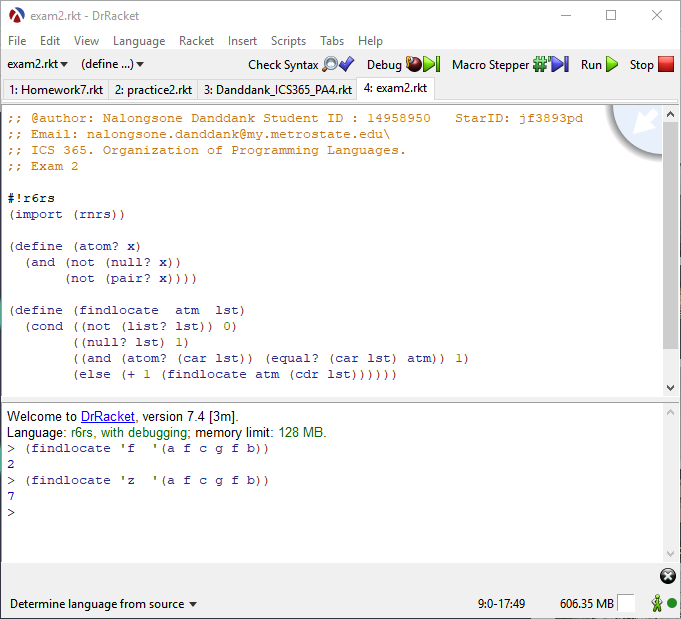
(define (findlocate atm lst)

(cond ((not (list? lst)) 0)

((null? lst) 1)

((and (atom? (car lst)) (equal? (car lst) atm)) 1)

(else (+ 1 (findlocate atm (cdr lst))))))



**Problem 5. 10 Points**

Consider the following program, written in JavaScript-like syntax:

// main program

**var** x, y, z;

**function** sub1() {

**var** a, y, z;

. . .

}

**function** sub2() {

**var** a, b, z;

. . .

}

**function** sub3() {

**var** a, x, w;

. . .

}

Given the following calling sequences and assuming that dynamic scoping

is used, what variables are visible during execution of the last subprogram

activated? Include with each visible variable the name of the unit where it is declared.

Example: main calls sub3; sub3 calls sub2; sub2 calls sub1 -

a, y, z sub1

b sub2

x, w sub3

Solution:

1. main calls sub1; sub1 calls sub2; sub2 calls sub3.

a, x, w sub3

b, z sub2

y sub1

1. main calls sub3; sub3 calls sub1.

a, y, z sub1

x, w sub3

1. main calls sub2; sub2 calls sub3; sub3 calls sub1

a, y, z sub1

x, w sub3

b sub2

1. main calls sub1; sub1 calls sub3.

a, x, w sub3

y, z sub1

1. main calls sub1; sub1 calls sub3; sub3 calls sub2.

a, b, z sub2

x, w sub3

y sub1