1. Email *

2. Full name *

3. What is the late policy for homeworks? *
   - Late work will never be accepted
   - Late work will not be accepted, but you have one 24h extension during the term
   - Late work will not be accepted, but you have two 24h extensions during the term
   - 10% off for each day late

4. What is the number one rule about the Honor Code (in any of your courses)? *
   - You may never do a web search for any of your assignments
   - You may not discuss the assignments with other students
   - If you are uncertain about how the Honor Code applies, ask your professor

5. We will cover four ________ in this course *
   - programming languages
   - programming paradigms
   - formal models of computation

6. Which linguistic hypothesis was covered in this lecture? *
   - The Church-Turing thesis
   - Chomsky's theory of universal grammar
   - The Sapir-Whorf hypothesis

7. What do you need to add to the end of the course page URL to access the reading materials? *
   - text
   - book
   - textbook
   - reading

This content is neither created nor endorsed by Google.
1. Email *

2. Full name *

3. What are Syntax and Semantics? *
   - Syntax is spelling, Semantics is grammar
   - Syntax is grammar, Semantics is spelling
   - Syntax is meaning, Semantics is grammar/spelling
   - Syntax is grammar/spelling, Semantics is meaning

4. Valid syntax for a programming language is usually specified using *
   - regular expressions
   - a context-free grammar
   - a clear description in English

5. Algorithmic checking of syntax / semantics *
   Careful, this one is tricky and was only briefly mentioned in the video - if you’ve taken CS 301 it might help you.
   - Mark only one oval.
     - Correct syntax of a program can be checked algorithmically
     - Correct semantics of a program can be checked algorithmically
     - Both of the first two statements are true
     - Neither of the first two statements are true

6. In the following BNF rule, what are the terminals? `<stmt>` ::= `<variable>` = `<expr>` ; *
   - Mark only one oval.
     - `<stmt>`
     - `<variable>` `<expr>`
     - `:=`
     - `;`

7. What is `(1 - 2) * 3` in prefix? *
   - Also tricky! Hint: visualize the expression tree.
   - Mark only one oval.
     - `- 1 2 * 3`
     - `* 3 - 1 2`
     - `1 2 - 3 *`
     - `* - 1 2 3`
CS 313 Quiz 3

Match 10 questions to 10 answers (one-to-one mapping)

1. First language aimed at liberal arts students
2. First functional language, based on lambda calculus
3. Which language? Multiply A by B giving C
4. Considered to be the first computer programmer
6. Used Backus-Naur form for publication of language description in 1960
7. Developed the first compiler in 1952; helped design COBOL
8. Designed in 1972 for coding UNIX routines
9. Oldest successful high-level language, designed for efficiently translating mathematical formulas
10. 20th century mathematician, pioneer of the modern digital computer architecture

1 - Grace Hopper
2 - COBOL
3 - C
4 - ALGOL
5 - FORTRAN
6 - John von Neumann
7 - LISP
8 - BASIC
9 - Ada
10 - Ada Byron Lovelace
1. Email *

2. What is the first and last word in a valid Pascal program? *
Mark only one oval.
- begin ...
- begin ...
- program ...
- program ...

3. Which is NOT a valid comment in Pascal? *
Mark only one oval.
- /* comment */
- (* comment *)
- { comment }

4. Is Pascal case sensitive? *
Mark only one oval.
- yes: temp, Temp, TEMP are all considered different
- no: capitalization of letters does not matter

5. Which is NOT a valid assignment operator in Pascal? *
Mark only one oval.
- =
- :=
- ::=*
- ==

6. Which is NOT a basic data type in Pascal? *
Mark only one oval.
- integer
- real
- double
- char
- Boolean

7. Which is NOT a valid if statement in Pascal? *
Mark only one oval.
- if x < 3 then y := 1;
- if (x < 3) then y := 1;
- if x < 3 then y := 1 else y := 0;
- if x < 3 then y := 1; else y := 0;

8. Which is NOT a valid loop in Pascal? *
Mark only one oval.
- repeat .... until ...
- loop .... while ...
- while .... do ...
- for .... do ...

9. What is the difference between a procedure and a function? *
Mark only one oval.
- none (two names for the same thing)
- a function has a return value, a procedure doesn't
- only functions can be recursive

10. Which of the following is NOT a valid parameter list? *
Mark only one oval.
- procedure p(x : integer; y : real);
- procedure p(x, y : integer);
- procedure p(x : integer; var y : integer);
- procedure p(x, y : var);

11. How do you return the value 7 from a function named f? *
Mark only one oval.
- return 7;
- retweet 7;
- f := 7;
1. Email *

2. Which is NOT a Pascal data type? *
   - record
   - dictionary
   - pointer
   - subrange
   - array
   - enumeration

3. What is the role of the semicolon in Pascal? *
   - statement terminator (required after each statement)
   - statement separator (used to separate multiple statements)

4. Suppose a procedure "doit" takes no parameters. How would we call this function? *
   - Mark only one oval.
     - doit();
     - doit;

5. Why is it generally a good idea to pass arrays by reference (precede the parameter with "var")? *
   - Mark only one oval.
     - It is faster, since no copy is made of the array
     - It allows the called function to modify the (caller's) array values
     - Both of the above

6. How can we increment an integer variable i? *
   - Mark only one oval.
     - inc();
     - i += 1;
     - i++;

7. Pascal allows defining nested procedures / functions *
   - Mark only one oval.

8. The following is a legal Pascal type declaration: type lr = (left, right); a = array[lr] of integer; *
   - Mark only one oval.
     - True
     - False

9. Which symbol denotes a pointer type in Pascal? *
   - Mark only one oval.
     - ^
     - &
     - *

10. Which symbol denotes "not equal" in Pascal? *
    - Mark only one oval.
     - !=
     - <=
     - >=

11. What is the null pointer in Pascal? *
    - Mark only one oval.
     - null
     - nil
     - 0
1. Email *

2. How do we declare p to be a pointer to an integer? *
   Mark only one oval.
   - var p : ^integer;
   - var ^p : integer;
   - var p^ : integer;
   - this is not possible in standard Pascal

3. How do we dereference a pointer p to print the value it points to? *
   Mark only one oval.
   - writeln(p);
   - writeln(p^);
   - writeln(^p);
   - this is not possible in standard Pascal

4. Address-of operator *
   How do we get a pointer p to hold the address of an integer variable x?
   Mark only one oval.
   - p := x^;
   - p := &x;
   - p := @x;
   - this is not possible in standard Pascal

5. In a program using a pointer p, the statement "p := nil;" can never create garbage (i.e. inaccessible memory cells) *
   Mark only one oval.
   - true
   - false

6. In a program using a pointer p, the statement "new(p);" can never create garbage (i.e. inaccessible memory cells) *
   Mark only one oval.
   - true
   - false

7. Unlike in Java (which has a garbage collector), in Pascal it is the responsibility of the programmer to dispose of dynamically allocated memory cells that are no longer used *
   Mark only one oval.
   - true
   - false

8. Given the definition "procedure q(var x : real); begin x:=0 end;" the call "q(t);" can change t's value (where t is a variable) *
   Mark only one oval.
   - true
   - false

9. A procedure can have some parameters that are passed by value, and others that are passed by reference *
   Mark only one oval.
   - true
   - false

10. Suppose we implemented our own version of Pascal's function "inc(x)" to increment an integer variable. How would we declare the parameter x? *
    Mark only one oval.
    - pass by reference (with "var")
    - pass by value (no "var")

11. The "inc(x)" function (where x is an int) could also be implemented in Java *
    Mark only one oval.
    - true
    - false
CS 313 Parameter Passing Quiz

1. Consider the following procedure definition and statements (in pseudo-code):

```pseudo-code
procedure proc1(p, q) {
    p := 1;
    q := 2;
}

i := 0;   a[0] := 0;   a[1] := 0;
proc1(i, a[i]);
```

For each of the following parameter-passing mechanisms, list the values of i, a[0], and a[1] after the above code has been executed.

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>a[0]</th>
<th>a[1]</th>
</tr>
</thead>
<tbody>
<tr>
<td>pass-by-value:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pass-by-reference:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pass-by-name / macro expansion:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Consider the following procedure definition and statements (in pseudo-code):

```pseudo-code
var c, d;       // global variables

procedure proc2(p, q) {
    c := p * 2;
    q := c - 1;
}

c := 3;   d := 2;
proc2(c, d);
```

For each of the following parameter-passing mechanisms, list the values of c and d after the above code has been executed.

<table>
<thead>
<tr>
<th></th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>pass-by-value:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pass-by-reference:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pass-by-value-result:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Email *

2. Which is NOT a C data type? *
   - int
   - double
   - struct
   - subrange
   - pointer

3. What is the role of the semicolon in C? *
   - statement terminator (required after each statement)
   - statement separator (used to separate multiple statements)

4. Suppose a void function "doit" takes no parameters. How would we call this function in C? *
   - doit();
   - doit;

5. Pass by reference *
   - C does not have pass-by-reference parameters, but a similar effect can be achieved using pointers and the "address of" operator &
   - C has pass-by-reference parameters prefixed by '&'

6. "if ((x = y) == z)" is a valid statement in C *
   - True
   - False

7. How do sizeof(int *) and sizeof(char *) relate? *
   - sizeof(int *) is larger
   - sizeof(char *) is larger
   - they are equal

8. Pointer arithmetic: what is printed by the following C code? char *s = "quiz";
   printf("%c", *(s + 3)); *
   - i
   - z
   - 0 (the number zero)
   - nothing – this code won’t compile

9. C does not have Booleans, it uses integers instead *
   - True
   - False

10. What C function can be used to get user input (like Pascal’s readln)? *
    - input
    - scanf
    - fread

11. C has both "while (...) { ... }" and "do { ... } while (...)" loops *
    - True
    - False
**CS 313 - Quiz 09 - C structs, stack frame, survey**

Please complete this quiz after reviewing the lecture materials and videos. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

---

1. Email *

2. Where are local variables and parameters stored? *
   - Mark only one oval.
     - In global memory
     - In the stack
     - In the heap

3. Where does "malloc" allocate memory? *
   - Mark only one oval.
     - In global memory
     - In the stack
     - In the heap

---

**Matching**

For each of the following 5 items, select the matching (equal / most similar) term from the 7 options given. Each option is used at most once.

4. activation record *
   - Mark only one oval.
     - access link
     - control link
     - struct
     - stack frame
     - *(a+i)
     - *(*a).i
     - *(a+i)

5. record *
   - Mark only one oval.
     - access link
     - control link
     - struct
     - stack frame
     - *(a+i)
     - *(*a).i
     - *(a+i)

---

6. old FP (frame pointer) *
   - Mark only one oval.
     - access link
     - control link
     - struct
     - stack frame
     - *(a+i)
     - *(*a).i
     - *(a+i)

7. a->i *
   - Mark only one oval.
     - access link
     - control link
     - struct
     - stack frame
     - *(a+i)
     - *(*a).i
     - *(a+i)

---

**Brief course survey**

Please answer the questions below to give me some feedback about the course so far. (All answers will receive full points once I manually assign them.)

8. a[] *
   - Mark only one oval.
     - access link
     - control link
     - struct
     - stack frame
     - *(a+i)
     - *(*a).i
     - *(a+i)

9. Overall *
   - How is the course going so far? (1 - terrible, 2 - bad, 3 - ok, 4 - good, 5 - great)
   - Mark only one oval.
     - 1
     - 2
     - 3
     - 4
     - 5

10. Pace *
    - How is the pace? (1 - much too slow, 2 - slow, 3 - good, 4 - fast, 5 - much too fast)
    - Mark only one oval.
      - 1
      - 2
      - 3
      - 4
      - 5
11. **Workload** *  
  How is the workload? (1 - very low, 2 - low, 3 - good, 4 - high, 5 - very high)  
  Mark only one oval.  

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>low</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>high</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12. Is there anything I should change about the course?  
  (optional) – Please let me know if there are aspects of the course that need improving.
1. Email *

2. What analogy does Steve Jobs use to explain OO programming? *
   - doing Laundry in San Francisco
   - buying Ben and Jerry's in Vermont
   - building a spaceship in Cupertino

3. Which is NOT an OO language? *
   - Simula
   - LISP
   - Smalltalk
   - C++
   - Java

4. What's the difference between method and message? *
   - method is a function, message is a procedure
   - method is a procedure definition, message is a procedure call
   - none – they are synonyms

5. Inheritance - which is NOT true? *
   - If A is a subclass of B, then B is a superclass of A
   - A subclass can override (redefine) a method declared in a superclass
   - Subclasses can add instance variables to those defined in a superclass
   - Subclasses can remove instance variables defined in a superclass

6. What does the Smalltalk expression $5 - 2 \times 2$ evaluate to? *
   - 1
   - 5
   - 6
   - 9
CS 313 - Quiz 11 - Smalltalk

Please complete this quiz after reviewing the lecture materials and videos. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

The respondent's email (null) was recorded on submission of this form.

* Required

1. Email *

2. What does this code do: #(2 4 6 8) at: 2 put: 3 *

Mark only one oval.

- It changes the array to #(3 4 6 8)
- It changes the array to #(2 3 6 8)
- It changes the array to #(2 4 3 8)
- It throws an error

3. Smalltalk has unary, binary, and keyword messages. What is their order of precedence (highest to lowest)? *

Mark only one oval.

- unary, binary, keyword
- unary, keyword, binary
- keyword, unary, binary
- keyword, binary, unary

4. What is a block? *

Mark only one oval.

- a keyword message with one argument
- the value returned from a method
- a group of Smalltalk statements, to be evaluated later

5. How does one write an if-statement in Smalltalk? *

Mark only one oval.

- by sending block(s) to a Boolean object via a keyword message
- by sending Boolean value(s) to a block
- using if:elif:else:

6. What does this code evaluate to? s := 0. (2 to: 4) do: [:x | s := s + x]. s. *

Mark only one oval.

- 0
- 4
- 6
- 9
- nil
- it throws an error

This content is neither created nor endorsed by Google.
Please complete this quiz after reviewing the lecture materials and videos. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web. You may interact with Squeak if you wish.

The respondent's email (null) was recorded on submission of this form.

* Required

1. Email *

2. Which of the following Smalltalk expressions yields the largest result? *
   Mark only one oval.
   - $(7 + 9) \sqrt{\text{max}: 5}$
   - $7 + (9 \sqrt{\text{max}: 5})$

3. Suppose b is a block with no parameters that prints 'Hello World' to the Transcript. How do we run the block? *
   Mark only one oval.
   - b doIt
   - b run
   - b value

4. Short-circuiting: Which of the following expressions properly ensures a valid array index > 0? *
   Mark only one oval.
   - $i > 0 \land (\text{a at: } i) = 7$
   - $(i > 0) \land (\text{a at: } i) = 7$
   - $i > 0 \land [(\text{a at: } i) = 7]$

5. Which variable names indicate the correct types for the do: method? *
   Mark only one oval.
   - aBlock do: aBoolean
   - aBlock do: anInteger
   - aCollection do: aBlock

6. How can we write the body of the "max: x" method? *
   Mark only one oval.
   - $x > \text{self ifTrue: }[^ x] \text{ifFalse: }[^ \text{self}]$
   - $\text{self > x ifTrue: }[^ \text{self}] \ast ^ x$
   - either of the first two options will work
   - neither of the first two options will work

This content is neither created nor endorsed by Google.
Implementing new classes

Here are some questions about our first example of defining a new class.

2. **1 point**
   Mark only one oval.
   - Stack
   - Queue
   - Vector
   - Linked List

3. **1 point**
   Mark only one oval.
   - a
   - c
   - q
   - data
   - self

5. **1 point**
   Mark only one oval.
   - #1: It returns an instance of the superclass
   - #2: It sends the "init" message to the superclass
   - #3: It sends the "new" message to the superclass
   - #4: It sends the "init" message to the new instance, then returns it
   - #2 followed by #1
   - #3 followed by #4

Survey question

(any answer will earn a point)

6. **1 point**
   Mark only one oval.
   - Prefer flipped classroom (like the first 3 weeks)
   - Prefer a mix, e.g. one in-person class per week (like we did this week)
   - Prefer all in-person classes
   - Other: ___________________________________________
1. Email *

2. In our IntList implementation, what is the total number of objects representing a list containing 3 integers? *
   Mark only one oval.
   - 1
   - 3
   - 4
   - 6

3. In the IntListElt class, what is "val"? *
   Mark only one oval.
   - an instance variable
   - an instance method
   - both

4. What are "setters" and "getters"? *
   Mark only one oval.
   - instance methods providing access to instance variables
   - class methods to access superclass and subclass
   - special types of blocks

5. In Smalltalk, all instance variables are private *
   Mark only one oval.
   - True
   - False

6. In the "do:" method in the IntListElt class, what does the statement "aBlock value: val" accomplish? *
   Mark only one oval.
   - It calls aBlock recursively on the rest of the list
   - It evaluates aBlock and stores the result in val
   - It evaluates aBlock with val as the parameter
   - It checks if aBlock == val

This content is neither created nor endorsed by Google.
1. How do you write hello world in Ruby? *

Mark only one oval.

- write("hello world")
- puts "hello world"
- "hello world".print

2. How do you add two numbers a and b? *

Mark only one oval.

- a + b
- a + (b)
- either of the above two options works

3. What do lists / arrays look like in Ruby? *

Mark only one oval.

- [10, 20, 30]
- (10, 20, 30)
- Ruby has no lists

4. What does assignment look like in Ruby? *

Mark only one oval.

- x <- 3
- x := 3
- x = 3

5. What is a dictionary called in Ruby? *

Mark only one oval.

- Map
- Hash
- Ruby doesn’t have dictionaries

6. Like Smalltalk, in Ruby everything is an object *

Mark only one oval.

- True
- False

7. How do you define a function f that returns the square of a number in Ruby? *

Mark only one oval.

- fun f(x): return x * x
- def f(x) x * x end

8. Like Smalltalk, Ruby has blocks *

Mark only one oval.

- True
- False

9. If x = "Daniel", how do we get the string "Hello Daniel, hi"? *

Mark only one oval.

- "Hello 'a, hi" + x
- "Hello", x, " hi"
- "Hello #t(x), hi"
1. Email *

2. How do you add new methods to an existing class like Integer? *

Mark only one oval.
- you “open” the class and provide the method definition using Ruby’s System Browser
- this is not possible in Ruby

3. What does (1..5).to_a evaluate to? *

Mark only one oval.
- [1, 2, 3, 4]
- [0, 1, 2, 3, 4]
- [1, 2, 3, 4, 5]

4. What is the result? a = [10, 20, 30]; a[1] *

Mark only one oval.
- 10
- 20
- nil

5. What is the result? x = ['a', 'b', 'c']; x.join('a') *

Mark only one oval.
- “a+b+c”
- “a b c”
- “abc’”
- “abc+”

6. How do you define a subclass Circle of a class Oval? *

Mark only one oval.
- class Circle subclass Oval
- class Circle extends Oval
- class Circle < Oval

7. What does it mean if a method name ends with a question mark? *

Mark only one oval.
- it takes optional parameters
- it uses randomization
- it returns a Boolean

8. What does a variable name starting with ‘@’ indicate? *

Mark only one oval.
- a global variable
- an instance variable
- a block

9. What does “attr_accessor :x” do in a class definition? *

Mark only one oval.
- it creates an instance variable @x
- it creates a “getter” method x
- it creates a “setter” method x=
- all of the above

10. What does “end” close in Ruby? *

Mark only one oval.
- class definitions
- function definitions
- loops
- all of the above

11. Popularity contest *

Which of the languages we’ve studied so far is your favorite? (No wrong answer :)

Mark only one oval.
- Pascal
- C
- Smalltalk
- Ruby
CS 313 - Quiz 18 - Functional Programming

Please complete this quiz after reviewing the lecture materials. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

The respondent's email (null) was recorded on submission of this form. * Required

1. Email *

2. What does Referential Transparency mean in a programming language? *
   Mark only one oval.
   - type safety of pointers
   - using descriptive variable names
   - the same expression has the same value everywhere

3. Why are functional languages relevant today? *
   Mark only one oval.
   - they best reflect the underlying von Neumann architecture
   - they allow easy parallelization, unlike imperative languages
   - they are easiest to translate into machine language

4. Which is not allowed in a pure functional language? *
   Mark only one oval.
   - duplicate function calls: y = f(x) + f(x)
   - incrementing a variable: x = x + 1
   - passing functions as parameters to other functions

5. How do you compute 3 + 4 in Scheme? *
   Mark only one oval.
   - (3 + 4)
   - 3 + (4)
   - (3 4 +)
   - (+ 3 4)

6. What does the Scheme expression (if #f 3 4) evaluate to? *
   Mark only one oval.
   - #f
   - #t
   - 3
   - 4

This content is neither created nor endorsed by Google.
1. Email *

2. Just like in Smalltalk everything is an object, in Scheme everything is a list *
   Mark only one oval.
   - True
   - False

3. What are more intuitive names for car, cdr? *
   Mark only one oval.
   - first, last
   - first, second
   - first, rest

4. How does Scheme display the list '(2 . (4 . (6 . ())))? *
   Mark only one oval.
   - '(2 4 6)
   - '(2 (4 (6)))
   - '(2 4 6 ()
   - '(2 4 . 6)

5. What is (car (cdr x))? *
   Mark only one oval.
   - a
   - b
   - c
   - d
   - (a)
   - (b)
   - (c)
   - (d)
   - (a b)
   - (c d)
   - ((a b))
   - ((c d))

6. What is (cdr (car x))? *
   Mark only one oval.
   - a
   - b
   - c
   - d
   - (a)
   - (b)
   - (c)
   - (d)
   - (a b)
   - (c d)
   - ((a b))
   - ((c d))

This content is neither created nor endorsed by Google.
1. Email *

2. What would (count-atoms '(a (b (c d)))) return? *

   Mark only one oval.
   
   ☐ 1
   ☐ 2
   ☐ 3
   ☐ 4

3. What would (reverse '(a (b (c d)))) return? *

   Mark only one oval.
   
   ☐ '(d (c b a))
   ☐ '(((c d) b) a)
   ☐ '(b (c d)) a
   ☐ none of the above options

4. Given a list x of length M and a list y of length N, what are both runtime and memory needs of (append x y)? *

   Mark only one oval.
   
   ☐ O(M)
   ☐ O(N)
   ☐ O(M+N)
   ☐ O(M*N)

5. Which of the following functions is tail-recursive? *

   (using a made-up syntax here)

   Mark only one oval.
   
   ☐ def f(x, y): if x<=0 return y else return 2 * f(x-1, y)
   ☐ def f(x, y): if x<=0 return y else return f(x-1, 2 * y)
   ☐ def f(x, y): if x<=0 return y else return f(x-1, y) + f(x-1, y)

6. What are time and space needs for the tail-recursive Scheme function to compute the N-th Fibonacci number? *

   Mark only one oval.
   
   ☐ O(1) time, O(1) space
   ☐ O(N) time, O(1) space
   ☐ O(N) time, O(N) space
   ☐ O(2^N) time, O(N) space

---

This content is neither created nor endorsed by Google.
Please complete this quiz after reviewing the lecture materials. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

The respondent's email (null) was recorded on submission of this form.

1. Email *

For the next 4 questions, suppose x is '((a b) (c d))

By the way, you are allowed to use DrRacket while you answer this quiz. But use your head first!

2. What is (length x)? *

Mark only one oval.

- 1
- 2
- 3
- 4

3. What expression returns 'a? *

Mark only one oval.

- (caar x)
- (cadr x)
- (cdar x)
- (caadr x)
- (cadar x)

4. What expression returns 'b? *

Mark only one oval.

- (caar x)
- (cadr x)
- (cdar x)
- (caadr x)
- (cadar x)

5. What expression returns 'c? *

Mark only one oval.

- (caar x)
- (cadr x)
- (cdar x)
- (caadr x)
- (cadar x)

6. What does ((if (< 1 2) + *) 3 4) evaluate to? *

Mark only one oval.

- 3
- 4
- 7
- 12
- error

7. What does "higher-order" mean? *

Mark only one oval.

- We can pass functions as parameters
- We can return a new function from a function
- We can assign functions to variables
- All of the above

8. What is the equivalent of Scheme's lambda expression in Smalltalk? *

Mark only one oval.

- A class method
- An instance method
- A block

9. Which definition is equivalent to (define (g x) (+ x 3))? *

Mark only one oval.

- (define (g x) (lambda (+ x 3)))
- (define g (lambda (+ x 3)))
- (define g (lambda (x) (+ x 3)))

10. What does (map car '((1 a) (2 b) (3 c))) return? *

Mark only one oval.

- (1 a)
- (1 2 3)
- (a b c)

11. What does (numderiv square 0.00001) return? * where square is defined as (lambda (x) (* x x))

Mark only one oval.

- a function that approximates (* x x)
- a function that approximates (* 2 x)
- the square of 0.00001
- approximately 10

This content is neither created nor endorsed by Google.
1. Email *

2. What is 'let' used for? *
   - Mark only one oval.
   - it's another way to write an if statement
   - to quit out of a function
   - to define one or more local variables

3. What is the difference between let and let*? *
   - Mark only one oval.
   - let evaluates the expressions in parallel, while let* evaluates them sequentially
   - let evaluates the expressions sequentially, while let* evaluates them in parallel
   - there is no difference

4. What does (deriv 'x 'x) return? *
   - Mark only one oval.
   - 0
   - 1
   - 'x
   - error

5. What does (deriv 'z 'x) return? *
   - Mark only one oval.
   - 0
   - 1
   - 'x
   - 2
   - error

6. What does (deriv '(* x x) 'x) return? *
   - Mark only one oval.
   - a function that approximates (* 2 x)
   - a function that computes (* 2 x)
   - the list (* 2 x)

7. (Just because I'm curious :) at what speed do you usually watch the videos? *
   (Bonus point, no wrong answer :)
   - Mark only one oval.
   - 0.5x
   - 0.75x
   - 1x
   - 1.25x
   - 1.5x
   - 1.75x
   - 2x
1. Email *

2. Which function does not exist in Scheme? *
   - Mark only one oval.
   - map
   - filter
   - reduce
   - foldl
   - foldr

3. Which expression gives the result 10? *
   - Mark only one oval.
   - (map (lambda (x) (max x 0)) '(-3 -10 10 3))
   - (map (lambda (x) (> x 0)) '(-3 -10 10 3))
   - (filter (lambda (x) (> x 0)) '(-3 -10 10 3))
   - (foldl max 0 '(-3 -10 10 3))

4. Which expression gives the result '(10 3)? *
   - Mark only one oval.
   - (map (lambda (x) (max x 0)) '(-3 -10 10 3))
   - (map (lambda (x) (> x 0)) '(-3 -10 10 3))
   - (filter (lambda (x) (> x 0)) '(-3 -10 10 3))
   - (foldl max 0 '(-3 -10 10 3))

5. Which expression gives the result '(0 10 3)? *
   - Mark only one oval.
   - (map (lambda (x) (max x 0)) '(-3 -10 10 3))
   - (map (lambda (x) (> x 0)) '(-3 -10 10 3))
   - (filter (lambda (x) (> x 0)) '(-3 -10 10 3))
   - (foldl max 0 '(-3 -10 10 3))

6. In a purely functional language, which three parameter passing mechanism cannot be distinguished? *
   - Mark only one oval.
   - call by value, call by name, call by reference
   - call by name, call by reference, call by value-result
   - call by value, call by reference, call by value-result

7. Given the function (define (p) (p)), what happens when we call (p)? *
   - Mark only one oval.
   - stack overflow
   - infinite loop
   - error

8. In a language with call-by-value parameter passing, we could define a "short-circuit" OR operation as: (define (OR a b) (if a #t b)) *
   - (using Scheme syntax)
   - Mark only one oval.
   - true
   - false

9. What does the ML expression 2 + 3.0 evaluate to? *
   - Mark only one oval.
   - 5
   - 5.0
   - this will result in an error

10. What does the ML expression 2 + ~3 * 4 evaluate to? *
    - Mark only one oval.
    - 4
    - ~4
    - 10
    - ~10
    - 20

11. What is the type of an ML function defined as "fun add(x, y) = x + y;"? *
    - Mark only one oval.
    - int -> int
    - (int, int) -> int
    - (int, int) -> int + int
    - int * int -> int

12. (optional) Do you want my help finding a partner for HW 7?
    - Check all that apply.
    - Yes

---

This content is neither created nor endorsed by Google.
CS 313 - Quiz 24 - ML

Please complete this quiz after reviewing the lecture materials. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

The respondent's email (null) was recorded on submission of this form. * Required

1. Email *

Note: You may use the SML interpreter while answering this quiz. But as always, use your head first.

2. How do we concatenate two strings s and t? *

Mark only one oval.

- s t
- s + t
- s * t
- concat(s, t)

3. Which results in a type error? *

Mark only one oval.

- 10 div 3
- 10.0 / 3.0
- round(10.0) / 3
- 10.0 / real(3)

4. What are ML’s names for AND and OR? *

Mark only one oval.

- and, or
- bothof, onefor
- also, else
- andalso, orelse

5. What is the type of (10, 22, 33.3)? *

Mark only one oval.

- int * int * real
- (int * int) * real
- int * (int * real)

6. If t = (10, 22, 33.3), how do we get the value 22? *

Mark only one oval.

- #1 t
- #2 t
- #1 (#2 t)
- #2 (#1 t)

7. What are ML’s names for Scheme’s functions car, cdr? *

Mark only one oval.

- first, rest
- fst, rst
- head, tail
- hd, tl

8. In ML, all elements in a list need to have the same type *

Mark only one oval.

- true
- false

9. Given the list x = [1, 22, 333], how do we get the value 22? *

Mark only one oval.

- hd (hd x)
- hd (tl x)
- tl (hd x)
- tl (tl x)

10. Given a = 1 and b = [2, 3], which is NOT a valid way to get [1, 2, 3]? *

Mark only one oval.

- a = b
- a @ b
- [a] @ b

11. What is the type of ML’s cons operator ::? *

Mark only one oval.

- 'a list -> 'a
- 'a list -> 'a list
- 'a * 'a list -> 'a list
- 'a list * 'a list -> 'a list

This content is neither created nor endorsed by Google.
1. **Email**
   * Required

   Note: You may use the SML interpreter while answering this quiz. But as always, use your head first.

2. **1 point**
   Mark only one oval.
   
   it allows defining local variables
   it allows defining local (helper) functions
   both of the above

3. **1 point**
   Mark only one oval.
   
   let
   let*
   lambda

4. **1 point**
   Similar to Python, we can "unpack" a tuple into multiple variables, e.g.: val t = (1, 2); val (a, b) = t; *
   Mark only one oval.
   
   True
   False

5. **1 point**
   Mark only one oval.
   
   copy a list
   reverse a list
   compute the length of a list
   sum the elements in a list

6. **1 point**
   Mark only one oval.
   
   True
   False

---

This content is neither created nor endorsed by Google.
For the following 5 questions, select the correct type for each function

2. fun neg n = 0 - n; * 1 point
   Mark only one oval.
   - fn : int -> int
   - fn : int -> unit
   - fn : unit -> int
   - fn : 'a -> 'a
   - fn : int -> int * int
   - fn : int -> int -> int
   - fn : int * int -> int
   - fn : int * int * int

3. fun id n = n; * 1 point
   Mark only one oval.
   - fn : int -> int
   - fn : int -> unit
   - fn : unit -> int
   - fn : 'a -> 'a
   - fn : int -> int * int
   - fn : int -> int -> int
   - fn : int * int -> int

4. fun predsucc n = (n - 1, n + 1); * 1 point
   Mark only one oval.
   - fn : int -> int
   - fn : int -> unit
   - fn : unit -> int
   - fn : 'a -> 'a
   - fn : int -> int * int
   - fn : int -> int -> int
   - fn : int * int -> int
   - fn : int * int * int

5. fun add x y = x + y; * 1 point
   Mark only one oval.
   - fn : int -> int
   - fn : int -> unit
   - fn : unit -> int
   - fn : 'a -> 'a
   - fn : int -> int * int
   - fn : int -> int -> int
   - fn : int * int -> int
   - fn : int * int * int

6. fun ans () = 42; * 1 point
   Mark only one oval.
   - true
   - false

The next 5 questions are about this function definition:

```ml
fun exp x 0 = 1.0
| exp x n = x * exp x (n-1);
```
11. exp (3.0; 2) is a valid expression

Mark only one oval.

☐ true
☐ false
Functional programming in Python

2. Which of map, filter, reduce does NOT return a list? *
   - map
   - filter
   - reduce

3. Which Python 2 expression does NOT yield \['11', '22', '33'\]? *
   (recall that 'hello' + 'world' = 'helloworld')
   - `map(lambda x: x+x, ['1', '2', '3'])`
   - `map(lambda x: 11*x, [1, 2, 3])`
   - `map(str, [11, 22, 33])`

Logic Programming / Prolog

7. How does Kowalski define 'algorithm'? *
   - algorithm = facts + rules
   - algorithm = logic + control
   - algorithm = if + while

11. In a Prolog rule "P :- Q1, Q2, Q3, Q4." what do the commas mean? *
    - AND
    - OR

This content is neither created nor endorsed by Google.
1. Email *

2. Recall that a comma between subgoals means AND. How do we implement OR? *
   - by providing multiple matching rules
   - by grouping subgoals in a list

3. How do we compute 4 * 5? *
   - X = 4 * 5.
   - X is 4 * 5.
   - eval(4 * 5, X).

4. Pete’s Puzzle: how many puzzles are there? *
   - 4! = 24
   - 10! = 3628800
   - 4^10 = 1048576
   - 10^4 = 10000

5. Given a list X, how do we extract the head H and tail T of X? *
   - (H::T) = X.
   - [H, T] = X.
   - [H|T] = X.

6. Given the query `member(X, [1, 2, 3]).` how many solutions do we get for X? *
   - none
   - 1
   - 3
   - an infinite number
CS 313 - Quiz 29 - Prolog 3

Please complete this quiz after reviewing the lecture materials. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

The respondent’s email (null) was recorded on submission of this form.

* Required

1. Email *

2. Which of the following is NOT a built-in Prolog list function? *
   - append
   - length
   - subsets
   - reverse

3. What does the following Prolog predicate do? *

   mystery([], Y, Y).
   mystery([H|X], Y, [H|Z]) :- mystery(X, Y, Z).

   Mark only one oval.
   - checks membership in a list
   - appends two lists
   - reverses a list using tail recursion
   - removes an element from a list

4. What is the answer to the query "loop(10, X)."? *

   Mark only one oval.
   - X = 0.
   - X = 10.
   - X = 20.
   - false.

5. Is "loop" tail recursive? *

   Mark only one oval.
   - yes
   - no

6. Which BST function is discussed in Lecture 29? *

   Mark only one oval.
   - inserting a value into a BST
   - removing a value from a BST
   - testing membership in a BST
1. Email *

2. When removing the value of a node with two children in a BST, what value does it get replaced with? *

Mark only one oval.
- the value of the left child
- the value of the right child
- the right-most value in the left subtree
- the right-most value in the right subtree

3. What is a substitution $\sigma$? *

Mark only one oval.
- a function mapping terms to variables
- a function mapping variables to terms
- solving the subgoals for a given rule

For the next 5 questions, select the most general unifier of the two terms
You may use the Prolog interpreter. But as always, use your head first.

4. $a(b) = a(X)$ *

Mark only one oval.
- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

5. $a(c, X) = a(b, Y)$. *

Mark only one oval.
- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

6. $a(c, X) = a(Y, Y)$. *

Mark only one oval.
- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

7. $a(X, c) = a(Y, b)$. *

Mark only one oval.
- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

8. $a(X, Y) = a(Y, X)$. *

Mark only one oval.
- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

9. What is Prolog's control algorithm? *

Mark only one oval.
- breadth-first search of the goal tree
- depth-first search of the goal tree

10. Changing goal order *

Mark only one oval.
- permutes the branches of the search tree (so solutions are found in a different order)
- can change the shape (e.g., height) of the search tree
- has no effect on the search tree

11. Changing rule order *

Mark only one oval.
- permutes the branches of the search tree (so solutions are found in a different order)
- can change the shape (e.g., height) of the search tree
- has no effect on the search tree

This content is neither created nor endorsed by Google.
CS 313 - Quiz 31 - Prolog 5

Please complete this quiz after reviewing the lecture materials. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

1. Email *

2. What is the purpose of the cut "!" in Prolog? *
   - it prevents backtracking past the cut
   - it allows fast reload of a Prolog file
   - it is a variable containing the last answer computed

3. What do we call a cut that changes the solutions found? *
   - red cut
   - green cut
   - blue cut

4. Which is NOT a common use of the cut in Prolog? *
   - increase efficiency by eliminating redundant comparisons
   - reducing memory usage by precompiling rules
   - pruning the search tree to avoid unwanted solutions

5. What do the following two rules implement? *
   - hnm(A) :- A, !, fail.
   - hnm(_).

6. What are the solution(s) to the following Prolog query? *
   - member(X, [1, 2, 3]), !.

This content is neither created nor endorsed by Google.
CS 313 - Quiz 32 - Searching for solutions

Please complete this quiz after reviewing the lecture materials. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

The respondent's email (null) was recorded on submission of this form.

* Required

1. Email *

2. What is not allowed on HW 10? *
   - solving more than 2 problems
   - using languages from more than 2 paradigms
   - looking at code on the web

3. What strategy is commonly employed to solve puzzles using programs? *
   - breadth-first search (BFS)
   - depth-first search (DFS)
   - minimax search

4. Which is harder (requires keeping track of additional information during the search)? *
   - searching for a solution for a problem with a cyclic state space
   - searching for a solution for a problem with an acyclic state space

5. Which language allows the implementation of DFS in the fewest lines of code? *
   - Pascal
   - Scheme
   - Prolog

6. How many solutions are there to the Wolf, Goat, Cabbage puzzle? *
   - none
   - 1
   - 2

This content is neither created nor endorsed by Google.
CS 313 - Quiz 33 - Haskell

Please complete this quiz after reviewing the lecture materials. You may refer to the lecture materials (anything directly linked from the current lecture page) while taking the quiz but you may not talk to other students or search the web.

The respondent's email (null) was recorded on submission of this form.

1. Email *

______________

2. Haskell is... *

Mark only one oval.

- purely functional
- lazy
- statically typed
- all of the above

3. In Haskell, strings are simply lists of characters *

Mark only one oval.

- true
- false

4. In Haskell, all elements of a list need to have the same type *

Mark only one oval.

- true
- false

5. In Haskell, all elements of a tuple need to have the same type *

Mark only one oval.

- true
- false

6. What is the "cons" operator in Haskell? *

Mark only one oval.

- +
- ++
- :
- ::

7. What is the "append" operator in Haskell? *

Mark only one oval.

- +
- ++
- :
- ::

8. What are the "car" and "cdr" functions in Haskell? *

Mark only one oval.

- fst, snd
- head, tail
- init, last

9. How do we define a function that squares its argument? *(assuming this definition appears in a source file; it won’t work at the prompt)*

Mark only one oval.

- def square x = x * x
- fun square x = x * x
- square x = x * x

10. How can we extract the letter 'e' from the string "hello"? *

Mark only one oval.

- snd 'hello'
- tail 'hello'
- let _x_ = "hello" in x

11. Which of the following expressions does NOT result in [2, 4, 6, 8, 10]? *

Mark only one oval.

- [2..10]
- [2*x | x <- [1..5]]
- map (+2) [1,2,3,4,5]
- take 5 [2,4..]
1. In Haskell, which of the following are evaluated lazily? * 1 point
Mark only one oval.
- variables in a let expression
- parameters in a function call
- elements of a list
- all of the above

2. In Haskell, all functions are "curried" and allow partial instantiation * 1 point
Mark only one oval.
- true
- false

3. In Haskell, which of the following are evaluated lazily? * 1 point
Mark only one oval.
- variables in a let expression
- parameters in a function call
- elements of a list
- all of the above

4. What do backticks mean in Haskell, e.g.: `123 + 456`? * 1 point
Mark only one oval.
- they convert f into a number
- they convert f into a string
- they convert a function f into an infix operator

5. Why can't we implement super-reverse in Haskell? * 1 point
E.g. superreverse [[1, 2], 3, [4]] = [[4], 3, [2, 1]]
Mark only one oval.
- Haskell doesn't allow recursion
- Haskell doesn't allow lists of lists
- Haskell doesn't allow "inhomogeneous" lists where elements have different types

6. In hw6.hs we define listTail = flip drop. What would be an alternative definition? * 1 point
Mark only one oval.
- listTail x y = drop x y
- listTail x y = drop y x
- listTail x y = flip x

7. How can we extract the letter 'o' from "hello"? * 1 point
Mark only one oval.
- last "hello"
- head (reverse "hello")
- 'hello' !! 4
- all of the above

8. The $ operator in Haskell means "apply". How can we write the following expression without it? * 1 point
Mark only one oval.
- head $ drop 3 "abcdef"
- (head . drop 3) "abcdef"
- head (drop 3) "abcdef"

9. How do we define an anonymous square function in Haskell? * 1 point
Mark only one oval.
- (\ x -> x * x)
- (\ x -> x * x)
- (\ x -> x * x)

10. What does the following code implement? * 1 point
Mark only one oval.
- quicksort
- mergesort
- infinite list of primes
- infinite list of Fibonacci numbers

This content is neither created nor endorsed by Google.