CS 313 Quiz 1

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

2. We will cover 4 ________ in this course
   - Programming languages
   - Programming language paradigms
   - Formal models of computation

3. Which linguistic hypothesis was covered in this video?
   - The Church-Turing thesis
   - Chomsky’s theory of universal grammar
   - The Sapir-Whorf hypothesis

4. What word do you need to add to the end of the URL of the course home page to access the reading materials?

5. The sound quality in this video...
   - was perfect!
   - needs improvement!
1. Which are the top two languages you and your classmates are interested in?
- Swift and C#
- Scala and Haskell
- JavaScript and C++

2. Syntax and semantics of programming languages
- 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

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

4. In the following BNF rule, what are the terminals?
   \[
   \langle \text{stmt} \rangle ::= \langle \text{variable} \rangle = \langle \text{expr} \rangle ;
   \]
- \langle stmt \rangle
- \langle variable \rangle \ <expr>
- \langle ; \rangle
- \langle = \rangle

5. What is \((1 - 2) \times 3\) in prefix?
- \(- 1 2 * 3\)
- \(* 3 - 1 2\)
- \(1 2 - 3 *\)
- \(* - 1 2 3\)
CS 313 Quiz 3

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

_  First language aimed at liberal arts students  a - Grace Hopper
_  First functional language, based on lambda calculus  b - COBOL
c - C
d - ALGOL
e - FORTRAN
_  Which language? Multiply A by B giving C  f - John von Neumann
g - LISP
_  Considered to be the first computer programmer  h - BASIC
_  Developed for US Department of Defense, 1975-1985  i - Ada
_  Used Backus-Naur form for publication of language description in 1960  k - Ada Byron Lovelace
_  Developed the first compiler in 1952; helped design COBOL
_  Designed in 1972 for coding UNIX routines
_  Oldest successful high-level language, designed for efficiently translating mathematical formulas
_  20th century mathematician, pioneer of the modern digital computer architecture
CS 313 - Quiz 04 - Pascal 1

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.

* Required

1. Email address *

2. Full name *

3. Lab section *
   Mark only one oval.
   - X
   - Y
   - Z

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

5. Comments *
   What are valid comments in Pascal? Check all that apply.
   Check all that apply:
   - /* comment */
   - (* comment *)
   - { comment }

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

7. Assignment *
   What operator does Pascal use for assignment to variables?
   Mark only one oval.
   - =
   - :=
   - ::= 
   - ==

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

9. If statement *
   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;

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

11. Procedures and functions *
    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

12. Parameters *
    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);

13. Return value *
    How do you return the value 7 from a function named f?
    Mark only one oval.
    - return 7;
    - retweet 7;
    - f := 7;
CS 313 - Quiz 05 - Pascal 2

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.

* Required

1. Email address *

2. Full name *

3. Lab section *

   Mark only one oval.
   - X
   - Y
   - Z

4. Data types *

   Mark only one oval.
   - record
   - dictionary
   - pointer
   - subrange
   - array
   - enumeration

5. Semicolon *

   What is the role of the semicolon in Pascal?

   - statement terminator (required after each statement)
   - statement separator (used to separate multiple statements)

6. Parameterless procedures *

   Suppose a procedure "doit" takes no parameters. How would we call this function?

   Mark only one oval.
   - doit();
   - doit;

7. Passing array parameters *

   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

8. Incrementing *

   How can we increment an integer variable i?

   Mark only one oval.
   - inc(i);
   - i += 1;
   - i++;
   - i++;  

9. Nested procedures *

   Pascal allows defining nested procedures / functions

   Mark only one oval.
   - True
   - False

10. Types *

    "type lr = (left, right); a = array[lr] of integer;" is a legal type declaration

    Mark only one oval.
    - True
    - False

11. Pointers *

    Which symbol denotes a pointer type in Pascal?

    Mark only one oval.
    - *
    - &
    - +

12. Not equal *

    Which symbol denotes "not equal" in Pascal?

    Mark only one oval.
    - =
    - !=
    - <=

13. Null *

    What is the null pointer in Pascal?

    Mark only one oval.
    - null
    - nil
    - 0

This content is neither created nor endorsed by Google.
1. Declaring pointers *
   How do we declare p to be a pointer to an integer?

   - var p : integer;
   - var *p : integer;
   - var p^ : integer;
   - this is not possible in standard Pascal

2. Address-of operator *
   How do we get a pointer p to hold the address of an integer variable x?

   - p := x^;
   - p := &x;
   - p := @x;
   - this is not possible in standard Pascal

3. Garbage *
   In a program using a pointer p, the statement 'p := nil;' can never create garbage (i.e. inaccessible memory cells)

   - true
   - false

4. Collecting garbage *
   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

   - true
   - false

5. Dereferencing *
   How do we dereference a pointer p to print the value it points to?

   - writeln(*p);
   - writeln(p^);
   - writeln(^p);
   - this is not possible in standard Pascal

6. Parameter passing *
   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)

   - true
   - false

7. Parameter passing 2 *
   A procedure can have some parameters that are passed by value, and others that are passed by reference

   - 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>i</th>
<th>a[0]</th>
<th>a[1]</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-name / macro expansion:</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>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>pass-by-value:</td>
<td></td>
</tr>
<tr>
<td>pass-by-reference:</td>
<td></td>
</tr>
<tr>
<td>pass-by-value-result:</td>
<td></td>
</tr>
</tbody>
</table>
CS 313 - Quiz 08 - C

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.

* Required

1. **Email address** *

   __________________________________________________________

2. **Full name** *

   __________________________________________________________

3. **Lab section** *

   Mark only one oval.
   - X
   - Y
   - Z

4. **Data types** *

   which is NOT a C data type?

   Mark only one oval.
   - int
   - double
   - struct
   - subrange
   - pointer

5. **Semicolon** *

   What is the role of the semicolon in C?

   Mark only one oval.
   - statement terminator (required after each statement)
   - statement separator (used to separate multiple statements)

6. **Parameterless function** *

   Suppose a void function "doit" takes no parameters. How would we call this function in C?

   Mark only one oval.
   - doit();
   - doit;

7. **Var parameters** *

   Mark only one oval.
   - 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 '&'

8. **assignments** *

   "if ((x = y) == z)" is a valid statement in C

   Mark only one oval.
   - True
   - False

9. **sizeof** *

   How do sizeof(int *) and sizeof(char *) relate?

   Mark only one oval.
   - sizeof(int *) is larger
   - sizeof(char *) is larger
   - they are equal

10. **pointer arithmetic** *

     what is printed by the C code: char *s = "quiz"; printf("%c", *(s + 3));

     Mark only one oval.
     - i
     - z
     - 0 (the number zero)
     - nothing – this code won’t compile

11. **Booleans** *

     C does not have Booleans, it uses integers instead

     Mark only one oval.
     - True
     - False

12. **user input** *

     what function can be used to get user input (like Pascal’s readln)?

     Mark only one oval.
     - input
     - scanf
     - fread

13. **loops** *

     C has both "while (...) { ... }" and "do {...} while (...)" loops

     Mark only one oval.
     - True
     - False

This content is neither created nor endorsed by Google.
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.

* Required

1. Email address *

2. Full name *

3. Lab section *

   Mark only one oval.
   - X
   - Y
   - Z

Matching

For each of the following 5 items, select the matching (equal / most similar) term from the 7 options given

4. activation record *

   - access link
   - control link
   - struct
   - stack frame
   - *(a+i)
   - (*a).i
   - &(a)+i

5. record *

   - access link
   - control link
   - struct
   - stack frame
   - *(a+i)
   - (*a).i
   - &(a)+i

6. old FP (frame pointer) *

   - access link
   - control link
   - struct
   - stack frame
   - *(a+i)
   - (*a).i
   - &(a)+i

7. a->i *

   - access link
   - control link
   - struct
   - stack frame
   - *(a+i)
   - (*a).i
   - &(a)+i

8. a[i] *

   - access link
   - control link
   - struct
   - stack frame
   - *(a+i)
   - (*a).i
   - &(a)+i

9. Your questions *

   Please list your questions about the material. Write down at least 2 questions, each on a single line.
   Anything goes! If you don’t have any questions write down 2 observations about things you learned or found interesting. We will go over your questions during lab, where I’ll ask each of you to state one of your own questions. I will award the points for your questions manually later.

   __________________________
   __________________________
   __________________________

   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. :)

   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

   terible
   great
CS 313 - Quiz 10 - OO programming

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.

* Required

1. Email address *

2. Full name *

3. Lab section *
   Mark only one oval.
   X
   Y
   Z

4. What is OO *
   What analogy does Steve Jobs use to explain OO programming?
   Mark only one oval.
   doing Laundry in San Francisco
   buying Ben and Jerry’s in Vermont
   building a spaceship in Cupertino

5. OO languages *
   Which is NOT an OO language?
   Mark only one oval.
   Simula
   LISP
   Smalltalk
   C++
   Java

6. method vs message *
   What’s the difference?
   Mark only one oval.
   method is a function, message is a procedure
   method is a procedure definition, message is a procedure call
   none – they are synonyms

7. Inheritance *
   Which is NOT true?
   Mark only one oval.
   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

8. Smalltalk expressions *
   What does the expression 5 - 2 * 2 evaluate to?

This content is neither created nor endorsed by Google.
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.

* Required

1. Email address *

2. Full name *

3. Lab section *
   
   Mark only one oval.
   
   X
   Y
   Z

4. Arrays *

   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

5. Message precedence *

   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

6. 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

7. Conditionals *

   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:

8. do: *

   What does the following expression 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.
CS 313 - Quiz 12 - Smalltalk 2

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. Please enter your middlebury.edu email.

1. Email address *

2. Full name *

3. Lab section *

   Mark only one oval.
   - X
   - Y
   - Z

4. Your questions *
   Please list your questions about the material. Write down at least 2 questions, each on a single line. Anything goes! If you don’t have any questions write down 2 observations about things you learned or found interesting. We will go over your questions during lab, where I’ll ask each of you to state one of your own questions. I will award the points for your questions manually later.

5. Smalltalk expressions *
   Which of the following expressions yields the largest result?

   Mark only one oval.
   - \(7 + 9\) \(\text{sqrt} \max: 5\)
   - \((7 + 9) \text{sqrt} \max: 5\)
   - \(7 + (9 \text{sqrt} \max: 5)\)

6. Short-circuiting *
   Which of the following expressions properly ensures a valid array index > 0?

   Mark only one oval.
   - \(i > 0\) and: \((a \at: i) = 7\)
   - \([i > 0] \and: (a \at: i) = 7\)
   - \(i > 0\) and: \([a \at: i] = 7\)

7. Defining new methods *
   How can we write the body of the "\(\text{max: x}\)" method?

   Mark only one oval.
   - \(x > \self \iftrue: [^x] \iffalse: [^\self]\)
   - \(\self > x \iftrue: [^\self] \^x\)
   - 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

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

5. What data structure did we implement? *

Mark only one oval.
- Stack
- Queue
- Vector
- Linked List

6. What is the name of the instance variable? *

Mark only one oval.
- a
- c
- q
- data
- self

7. What is the type of the instance variable? *

Mark only one oval.
- Array
- OrderedCollection
- Boolean

8. What does " ^ super new init " do? *

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
CS 313 - Quiz 14 - Smalltalk 4

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.

* Required

1. Email address

2. Full name

3. Lab section

   Mark only one oval.
   - X
   - Y
   - Z

4. IntList

   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

5. IntListElt

   In the IntListElt class, what is “val”?

   Mark only one oval.
   - an instance variable
   - an instance method
   - both

6. 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

7. private / public

   In Smalltalk, all instance variables are private

   Mark only one oval.
   - True
   - False

8. value

   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
CS 313 - Quiz 16 - Ruby 1

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.

* Required

1. Email address *

2. Full name *

3. Lab section *
   - Mark only one oval.
   - X
   - Y
   - Z

4. Printing *
   How do you write hello world in Ruby?
   - Mark only one oval.
   - write("hello world")
   - puts "hello world"
   - "hello world".print

5. Method invocation *
   How do you ask a string object s for its length?
   - Mark only one oval.
   - s.length()
   - s.length
   - length(s)

6. Lists / Arrays *
   What do lists / arrays look like in Ruby?
   - Mark only one oval.
   - [10, 20, 30]
   - (10, 20, 30)
   - Ruby has no lists

7. Assignment *
   What does assignment look like in Ruby?
   - Mark only one oval.
   - x = 3
   - x := 3
   - x <- 3

8. Dictionaries *
   What is a dictionary called in Ruby?
   - Mark only one oval.
   - Map
   - Hash
   - Ruby doesn't have dictionaries

9. Like in Smalltalk, in Ruby everything is an object *
   - Mark only one oval.
   - True
   - False

10. Like Smalltalk, Ruby has blocks *
    - Mark only one oval.
    - True
    - False

11. Functions *
    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

12. If statements *
    What "closes" an if statement in Ruby?
    - Mark only one oval.
    - if ...
    - if ...
    - if ...

13. String formatting *
    If x = "Daniel", how do we get the string "Hello Daniel, hi"?
    - Mark only one oval.
    - "Hello %s, hi" + x
    - "Hello " . x . " hi"
    - "Hello #{x}, hi"

This content is neither created nor endorsed by Google.

Google Forms
1. Email address *

2. Full name *

3. Lab section *
   Mark only one oval.
   - X
   - Y
   - Z

4. 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

5. 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]

6. What is the result? a = [10, 20, 30]; a[1] *
   Mark only one oval.
   - 10
   - 20
   - nil

7. What is the result? x = ['a', 'b', 'c']; x.join('+') *
   Mark only one oval.
   - 'a+b+c'
   - 'a b c'
   - 'abc'
   - 'abc+'

8. 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

9. 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

10. What does a variable name starting with '@' indicate? *
    Mark only one oval.
    - a global variable
    - an instance variable
    - a block

11. 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

12. What does "end" close in Ruby? *
    Mark only one oval.
    - class definitions
    - function definitions
    - loops
    - all of the above

13. 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.

* Required

1. Email address *

__________________________________________

2. Full name *

__________________________________________

3. Lab section *
Mark only one oval.

☐ X
☐ Y
☐ Z

4. Referential Transparency *
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

5. 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

6. 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

7. How do you compute \( 3 + 4 \) in Scheme? *

Mark only one oval.

☐ \((3 + 4)\)
☐ \(3+(4)\)
☐ \((3 \ 4 +)\)
☐ \((+ \ 3 \ 4)\)

8. What does the Scheme expression \((\text{if} \ #f \ 3 \ 4)\) evaluate to? *

Mark only one oval.

☐ #t
☐ #f
☐ 3
☐ 4

This content is neither created nor endorsed by Google.

Google Forms
CS 313 - Quiz 19 - Scheme lists

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.

* Required

1. Email address *


2. Full name *


3. Lab section *

   Mark only one oval.
   - X
   - Y
   - Z

4. Just like in Smalltalk everything is an object, in Scheme everything is a list *

   Mark only one oval.
   - True
   - False

5. What are more intuitive names for car, cdr? *

   Mark only one oval.
   - first, last
   - first, second
   - first, rest

6. 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)

   For the next 2 questions, suppose x is '((a b) (c d))

7. 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))

8. 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.
CS 313 - Quiz 20 - More Scheme & tail recursion

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.

* Required

1. Email address *

2. Full name *

3. Lab section *
   - X
   - Y
   - Z

4. What would `(count-atoms '(a (b (c d))))` return? *
   - Mark only one oval.
   - 1
   - 2
   - 3
   - 4

5. 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

6. Given a list x of length N and a list y of length M, what are both runtime and memory needs of `(append x y)`? *
   - Mark only one oval.
   - O(N)
   - O(M)
   - O(N+M)
   - O(N*M)

7. 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)`

8. 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.

Google Forms
For the next 3 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!

5. What expression returns 'a? *
   Mark only one oval.
   - (caar x)
   - (cadr x)
   - (cdar x)
   - (caadr x)
   - (cadar x)

6. What expression returns 'b? *
   Mark only one oval.
   - (caar x)
   - (cadr x)
   - (cdar x)
   - (caadr x)
   - (cadar x)

7. What expression returns 'c? *
   Mark only one oval.
   - (caar x)
   - (cadr x)
   - (cdar x)
   - (caadr x)
   - (cadar x)

8. What does ((if (+ 1 2) *) 3 4) evaluate to? *
   Mark only one oval.
   - 3
   - 4
   - 7
   - 12
   - error

9. 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

10. 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 + 3))

11. What does (map car '((1 a) (2 b) (3 c)) return? *
    Mark only one oval.
    - '(1 a)
    - '(1 2 3)
    - '(a b c)

12. What does (numderiv square 0.00001) return? *
    Mark only one oval.
    - a function that approximates (* x x)
    - a function that approximates (* 2 x)
    - the square of 0.00001
    - approximately 10

8. 1 point
Mark only one oval.
3
4
7
12
error
9. 1 point
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
10. 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)))

For the next 3 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!

5. What expression returns 'a? *
   Mark only one oval.
   - (caar x)
   - (cadr x)
   - (cdar x)
   - (caadr x)
   - (cadar x)

6. What expression returns 'b? *
   Mark only one oval.
   - (caar x)
   - (cadr x)
   - (cdar x)
   - (caadr x)
   - (cadar x)

7. What expression returns 'c? *
   Mark only one oval.
   - (caar x)
   - (cadr x)
   - (cdar x)
   - (caadr x)
   - (cadar x)

8. 1 point
Mark only one oval.
3
4
7
12
error
9. 1 point
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
10. 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)))
CS 313 - Quiz 22 - let, let*, deriv

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.

* Required

1. Email address *

2. Full name *

3. Lab section *
Mark only one oval.
[ ] X
[ ] Y
[ ] Z

4. 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

5. 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

6. What does (deriv 17 'x) return? *
Mark only one oval.
[ ] 0
[ ] 1
[ ] 17
[ ] 'x

7. What does (deriv 'x 'x) return? *
Mark only one oval.
[ ] 0
[ ] 1
[ ] 17
[ ] 'x

8. What does (deriv square-expr 'x) return? *
where square-expr is defined as the list '(* x x)
Mark only one oval.
[ ] a function that approximates (* 2 x)
[ ] a function that computes (* 2 x)
[ ] the list '(* 2 x)

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

This content is neither created nor endorsed by Google.
CS 313 - Quiz 23 - Scheme, evaluation order, 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.

* Required

1. Email address *

2. Full name *

3. Lab section *
   - X
   - Y
   - Z

4. Which function does not exist in Scheme? *
   - map
   - filter
   - reduce
   - foldl
   - foldr

5. Which expression gives the result 10? *
   - (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. Which expression gives the result '10 3)? *
   - (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))

7. Which expression gives the result '(0 0 10 3)? *
   - (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))

8. In a purely functional language, which three parameter passing mechanism cannot be distinguished? *
   - 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

9. Given the function (define (p) (p)), what happens when we call (p)? *
   - stack overflow
   - infinite loop
   - error

10. 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)

11. What does the ML expression 2 + 3.0 evaluate to? *
    - 5
    - 5.0
    - this will result in an error

12. What does the ML expression 2 + -3 * 4 evaluate to? *
    - 4
    - -4
    - 10
    - -10
    - 20

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

14. (optional) Do you want my help finding a partner for HW 7?
    - Check all that apply:
    - Yes
1. Email address *

2. Full name *

3. Lab section *
   Mark only one oval.
   X
   Y
   Z

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

4. How do we concatenate two strings s and t? *
   Mark only one oval.
   s + t
   s ^ t
   concat(s, t)

5. Which results in a type error? *
   Mark only one oval.
   10 div 3
   10.0 / 3.0
   int(10.0) / 3
   10.0 / real(3)

6. What are ML's names for AND and OR? *
   Mark only one oval.
   and, or
   bothof, oneof
   also, else
   andalso, orelse

7. What is the type of ((10, 22), 33.3)? *
   Mark only one oval.
   int * int * real
   (int * int) * real
   int * (int * real)

8. 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)

9. What are ML's names for Scheme's functions car, cdr? *
   Mark only one oval.
   first, rest
   fst, rst
   head, tail
   hd, tl

10. In ML, all elements in a list need to have the same type *
    Mark only one oval.
    true
    false

11. 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)

12. 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

13. What is the type of ML's cons operator ::? *
    Mark only one oval.
    'a list -> 'a
    'a list -> 'a list
    'a :: 'a list = 'a
    'a list :: 'a list = 'a list

14. 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)
**CS 313 - Quiz 25/26 - 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.

* Required

1. Email address *

2. Full name *

3. Lab section *

Mark only one oval.

X

Y

Z

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

For the following 5 questions, select the correct type for each function

### 4. fun neg n = 0-n; *

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 |

### 5. fun id n = n; *

Mark only one oval.

| fn: int -> int |
| fn: int -> unit |
| fn: unit -> int |
| fn: 'a -> 'a |
| fn: int -> int * int |
| fn: int -> int |
| fn: int * int -> int |

### 6. fun predsucc n = (n-1, n+1); *

Mark only one oval.

| fn: int -> int |
| fn: int -> unit |
| fn: unit -> int |
| fn: 'a -> 'a |
| fn: int -> int * int |
| fn: int -> int |
| fn: int * int -> int |

### 7. fun add x y = x + y; *

Mark only one oval.

| fn: int -> int |
| fn: int -> unit |
| fn: unit -> int |
| fn: 'a -> 'a |
| fn: int -> int * int |
| fn: int -> int |
| fn: int * int -> int |

### 8. fun ans () = 42; *

Mark only one oval.

| fn: int -> int |
| fn: int -> unit |
| fn: unit -> int |
| fn: 'a -> 'a |
| fn: int -> int * int |
| fn: int -> int |
| fn: int * int -> int |

### 9. The function `exp` is a so-called 'Haskell function' *

Mark only one oval.

| true |
| false |

### 10. The function `exp` is a so-called 'Curried function' *

Mark only one oval.

| true |
| false |

### 11. exp 3.0; is a valid expression *

Mark only one oval.

| true |
| false |

### 12. exp (3.0, 2); is a valid expression *

Mark only one oval.

| true |
| false |

### 13. What is the type of this function? *

Mark only one oval.

| fn: (real * int) -> real |
| fn: (real -> int) -> real |
| fn: real -> (int -> real) |

The next 5 questions are about this function definition:

```sml
val exp = fn x 0 => 1.0
    | exp x n = x * exp (n-1);
```

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

2. Full name *

3. Lab section *
   Mark only one oval.
   X
   Y
   Z

4. Which of map, filter, reduce does NOT return a list? *
   Mark only one oval.
   map
   filter
   reduce

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

6. What does \([x^2 \text{ for } x \in [-2, -1, 0, 1, 2]}\) evaluate to? *
   Mark only one oval.
   [-2, -1, 0, 1]
   [1, 0, 1, 4]
   [4, 1, 0, 1, 4]
   [4, 1, 0, 1]
   [-4, -1, 0, 1]

7. if \(\text{mul} = \lambda x, y. x^2 y\), what is \(\text{partial(\text{mul}, 2)}\)? *
   assume we imported partial from functools in Python 3
   Mark only one oval.
   the same as lambda x: 2*x
   the same as lambda x: x^2
   the same as lambda x: 2

8. How can we compute 5 factorial? *
   Mark only one oval.
   map(lambda x: x*x, [1, 2, 3, 4, 5])
   filter(lambda x: x*x, [1, 2, 3, 4, 5])
   reduce(lambda x, y: x*y, [1, 2, 3, 4, 5])

9. How does Kowalski define ‘algorithm’? *
   Mark only one oval.
   algorithm = facts + rules
   algorithm = logic + control
   algorithm = if + while

10. What is not a part of Prolog programming? *
    Mark only one oval.
    facts
    rules
    theorems
    queries

11. Which of the following is a valid Prolog variable? *
    Mark only one oval.
    gold
    i
    Anna

12. Suppose a Prolog program only contains the rule "coin(penny)". Which query would result in "false"? *
    Mark only one oval.
    ? - coin(penny).
    ? - coin(dime).
    ? - coin(X).

13. In a Prolog rule "P :- Q1, Q2, Q3, Q4," what do the commas mean? *
    Mark only one oval.
    AND
    OR

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

2. Full name *

3. Lab section *
   Mark only one oval.
   
   - X
   - Y
   - Z

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

5. How do we compute $4 \times 5$? *
   Mark only one oval.
   
   - $X = 4 \times 5$
   - $X$ is $4 \times 5$
   - eval($4 \times 5$, X)

6. Pete’s Puzzle: how many puzzles are there? *
   Mark only one oval.
   
   - 4! = 24
   - 10! = 3628800
   - $4 \times 10 = 1048576$
   - $10 \times 4 = 10000$

7. Given a list X, how do we extract the head H and tail T of X? *
   Mark only one oval.
   
   - (H:T) = X
   - [H, T] = X
   - [H|T] > X

8. Given the query “member(X, [1, 2, 3]).” how many solutions do we get for X? *
   Mark only one oval.
   
   - 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.

* Required

1. Email address *

2. Full name *

3. Lab section *
   Mark only one oval.
   - X
   - Y
   - Z

4. Which of the following is NOT a built-in Prolog list function? *
   Mark only one oval.
   - append
   - length
   - subsets
   - reverse

5. What does the following Prolog predicate do? *
   - checks membership in a list
   - appends two lists
   - reverses a list using tail recursion
   - removes an element from a list

6. The next 2 questions are about the following Prolog predicate:
   \[
   \text{loop}(\emptyset, \emptyset), \\
   \text{loop}(A, B) :- \\
   \text{AA} \leftarrow A - 1, \\
   \text{loop}((\text{AA}, \text{BB}), \\
   B \leftarrow \text{BB} + 2.
   \]

   What does the following Prolog predicate do? *
   - checks membership in a list
   - appends two lists
   - reverses a list using tail recursion
   - removes an element from a list

   What is the answer to the query "\text{loop}(10, X)."? *
   - \(X = 0\)
   - \(X = 10\)
   - \(X = 20\)
   - \text{false.}

7. Is "\text{loop}" tail recursive? *
   Mark only one oval.
   - yes
   - no

8. 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

This content is neither created nor endorsed by Google.
CS 313 - Quiz 30 - Prolog 4

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.

* Required

1. Email address *


2. Full name *


3. Lab section *

Mark only one oval.

- X
- Y
- Z

4. 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

5. 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.

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

Mark only one oval.

- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

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

Mark only one oval.

- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

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

Mark only one oval.

- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

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

Mark only one oval.

- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

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

Mark only one oval.

- $X = b$
- $X = c$
- $X = Y$
- $X = b, Y = c$
- $X = c, Y = c$
- false (cannot be unified)

11. What is Prolog's control algorithm? *

Mark only one oval.

- breadth-first search of the goal tree
- depth-first search of the goal tree

12. What affects the order in which solutions are found? *

Mark only one oval.

- goal order
- rule order

13. A query resulting in infinite search could be caused by *

Mark only one oval.

- goal order
- rule order
- either goal order or rule order
1. Email address *

___________________________________________

2. Full name *

___________________________________________

3. Lab section *

Mark only one oval.

□ X
□ Y
□ Z

4. What is the purpose of the cut "!" in Prolog? *

Mark only one oval.

□ it prevents backtracking past the cut
□ it allows fast reload of a Prolog file
□ it is a variable containing the last answer computed

5. What do we call a cut that changes the solutions found? *

Mark only one oval.

□ red cut
□ green cut
□ blue cut

6. Which is NOT a common use of the cut in Prolog? *

Mark only one oval.

□ increase efficiency by eliminating redundant comparisons
□ reducing memory usage by precompiling rules
□ pruning the search tree to avoid unwanted solutions

7. What do the following two rules implement? *

Mark only one oval.

□ logical NOT
□ logical OR
□ logical AND

8. What are the solution(s) to the following Prolog query? *

Mark only one oval.

□ false.
□ X = 1.
□ X = 1; X = 2; X = 3.
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.

* Required

1. Email address *

2. Full name *

3. Lab section *

   Mark only one oval.
   - [ ] X
   - [ ] Y
   - [ ] Z

4. What is not allowed on HW 10? *

   Mark only one oval.
   - [ ] solving more than 2 problems
   - [ ] using languages from more than 2 paradigms
   - [ ] looking at code on the web

5. What strategy is commonly employed to solve puzzles using programs? *

   Mark only one oval.
   - [ ] breadth-first search (BFS)
   - [ ] depth-first search (DFS)
   - [ ] minimax search

6. Which is harder (requires keeping track of additional information during the search)? *

   Mark only one oval.
   - [ ] searching for a solution for a problem with a cyclic state space
   - [ ] searching for a solution for a problem with an acyclic state space

7. Which language allows the implementation of DFS in the fewest lines of code? *

   Mark only one oval.
   - [ ] Pascal
   - [ ] Scheme
   - [ ] Prolog

8. How many solutions are there to the Wolf, Goat, Cabbage puzzle? *

   Mark only one oval.
   - [ ] none
   - [ ] 1
   - [ ] 2
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.

* Required

1. Email address *

2. Full name *

3. Lab section *

   Mark only one oval.
   - X
   - Y
   - Z

4. Haskell is... *

   Mark only one oval.
   - purely functional
   - lazy
   - statically typed
   - all of the above

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

   Mark only one oval.
   - true
   - false

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

   Mark only one oval.
   - true
   - false

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

   Mark only one oval.
   - true
   - false

8. What is the “cons” operator in Haskell? *

   Mark only one oval.
   - +
   - ++
   - :
   - ::

9. What is the “append” operator in Haskell? *

   Mark only one oval.
   - +
   - ++
   - :
   - ::

10. 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..]

11. 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

12. How can we extract the letter ‘e’ from the string “hello”? *

    Mark only one oval.
    - snd "hello"
    - tail "hello"
    - let _x_ = "hello" in x

13. What are the “car” and “cdr” functions in Haskell? *

    Mark only one oval.
    - fst, snd
    - head, tail
    - init, last

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

    Mark only one oval.
    - true
    - false

15. 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..]
In Haskell, all functions are "curried" and allow partial instantiation.
Mark only one oval.
- true
- false

What do backticks mean in Haskell, e.g.: `123 * 456`?
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

Why can't we implement super-reverse in Haskell?
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

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

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

How do define an anonymous square function in Haskell?
Mark only one oval.
- (lambda x -> x * x)
- (fn x -> x * x)
- (\x -> x * x)

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

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