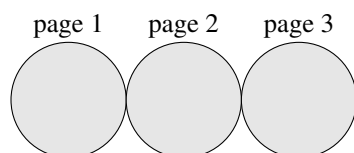
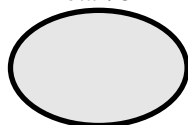


\$Id: cmps112-2011q2-exam1.mm,v 1.45 2011-04-22 19:59:45-07 - - \$



Total / 31



Please print clearly :

Name :

Login :

@ucsc.edu

No books ; No calculator ; No computer ; No email ; No internet ; No notes ; No phone. Neatness counts ! Do your scratch work elsewhere and enter only your final answer into the spaces provided.

1. Fill in the blank with a lambda expression so that `mul` behaves as shown here. [1✓]

The answer, in Perl, would be :

```
my $mul = sub {my $x = $_[0]; sub {$x * $_[0]}; print $mul->(3)->(4), "\n";
```

```
> (define mul (_____))
```

```
> ((mul 3) 4)
```

```
12
```

2. Fill in the following table. Answers (in alphabetical order) are: John Backus ; Grace Hopper *et.al.* ; John Kemeny & Thomas Kurtz ; John McCarthy. [1✓]

FORTRAN (1957)	LISP (1958)	COBOL (1959)	BASIC (1964)
----------------	-------------	--------------	--------------

3. Write an expression in Java using the operator `&&` which will return a result because `&&` is a lazy operator, but which would throw an exception if it were evaluated eagerly. [1✓]

4. Write a Perl program that will read `STDIN` and print the number of characters, words, and lines in the file. A word is any sequence of non-whitespace characters. [2✓]

```
% cat t
this is a test
second line
last line in the file
% wc.perl <t
      3      11      49
```

5. Define a function in Scheme whose first argument is a predicate (a function of one argument which returns either `#t` or `#f`) and whose second argument is a list. If the predicate returns `#t` for any element of the list, `contains` returns `#t`, otherwise it returns `#f`. [2✓]

```
> (contains (lambda (x) (= x 3)) '(1 2 3 4))
```

```
#t
```

```
> (contains (lambda (x) (> x 9)) '(1 2 3 4))
```

```
#f
```

6. Write a program in Perl which reads either the standard input or all of the files whose names are given on the command line and prints out each word followed by the number of times it appears in the file. Words are printed in lexicographic order. A word matches `\w+` and each word is printed only once. Your program must use `strict`; use `warnings`; [3✓]

```
% ./count.perl f1 f2
a 4
file 10
second 2
test 4
the 10
```

7. Define a Scheme function `map2` whose arguments are a binary function and two lists. It returns a list whose length is the same as the shorter list consisting of elements by using the function to combine corresponding elements of the list. [3✓]

```
> (map2 * '(1 2 3) '(4 5 6))
(4 10 18)
> (map2 = '(1 2 3) '(1 4 3 8 9))
(#t #f #t)
```

8. Define a function `foldl` which takes a function, a unit value, and a list, and returns a single value by applying the function between each member of the list. It must be tail recursive. [2✓]

```
> (foldl + 0 '(4 10 18))
27
> (foldl cons '() '(1 2 3))
(((()) . 1) . 2) . 3)
```

9. Define a function `ip` which computes the inner product of two lists by using `map2` and `foldl`. Elements of the two lists are pair-wise multiplied together, and the sum is taken. [1✓]

```
> (ip '(1 2 3) '(4 5 6))
32
```

10. Explain the difference between static and dynamic scope, and give an example of each. [2✓]

11. Draw a picture of the following Scheme expression. For each cell, draw a rectangular box with two fields in it, and draw a pointer from inside the box to the thing being pointed at. Atoms are drawn outside of all of these boxes. [2✓]

```
((a b c) (d ((e)) f 6))
```

Multiple choice. To the *left* of each question, write the letter that indicates your answer. Write **Z** if you don't want to risk a wrong answer. Wrong answers are worth negative points. **[11✓]**

number of correct answers		$\times 1 =$	$= a$
number of wrong answers		$\times \frac{1}{2} =$	$= b$
number of missing answers		$\times 0 =$	0
column total $c = \max(a - b, 0)$	11		$= c$

- Which of the following Java operators uses lazy evaluation ?
(A) `&&`
(B) `++`
(C) `==`
(D) `>>`
- Which Scheme expression will evaluate to the atom 2 ?
(A) `(caar ' (1 2 3))`
(B) `(cadr ' (1 2 3))`
(C) `(cdar ' (1 2 3))`
(D) `(cddr ' (1 2 3))`
- Which of the following is a higher-order function whose arguments are a function and a list, and whose result is a list containing the results of having applied the function to each element of the list ?
(A) `filter`
(B) `foldl`
(C) `lambda`
(D) `map`
- For a list of length n , `foldr`, which does not use tail recursion, will take up how much stack space ?
(A) $O(1)$
(B) $O(\log n)$
(C) $O(n)$
(D) $O(n \log n)$
- When is a static variable in C bound to a specific virtual address ?
(A) When the compiler produces an object file.
(B) When the linker produces an executable image.
(C) When the program is loaded via the `exec` system call.
(D) When the main function is called.
- A variable declared inside a C or Java function has :
(A) Dynamic scope and dynamic lifetime.
(B) Dynamic scope and static lifetime.
(C) Static scope and dynamic lifetime.
(D) Static scope and static lifetime.
- With respect to Java, the term “overloading” refers to :
(A) Automatic type conversion when the argument does not match the declared type of the parameter.
(B) Generic classes with type parameterization.
(C) Multiple functions with the same name and different signatures, defined in the same class.
(D) Multiple functions with the same name and signature, defined in different classes.
- Perl and Scheme are languages whose type checking is :
(A) Strong and dynamic.
(B) Strong and static.
(C) Weak and dynamic.
(D) Weak and static.
- After a Perl program has executed a subprocess, the completion status of that process is available in what variable ?
(A) `$!`
(B) `$&`
(C) `$?`
(D) `$_`
- Which expression will cause the empty list to be passed into `f` ?
(A) `(f '())`
(B) `(f ())`
(C) `(f null?)`
(D) `f (null)`
- Which of these languages was designed for use in business, finance, and administrative systems for companies and governments.
(A) BASIC
(B) COBOL
(C) FORTRAN
(D) LISP

* From <http://community.schemewiki.org/?scheme-fortune-cookies>:

“Around here, arguing that Java is better than C++ is like arguing that grasshoppers taste better than tree bark.” --- Thant Tessman <thant@acm.org> in comp.lang.scheme.

“Functional programming is like describing your problem to a mathematician. Imperative programming is like giving instructions to an idiot.” --- arcus, #scheme on Freenode.