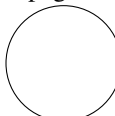
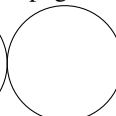
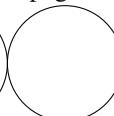
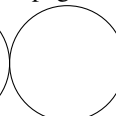



\$Id: cmps112-2017q2-midterm.mm,v 1.31 2017-05-04 13:47:18-07 - - \$

page 1	page 2	page 3	page 4	Total / 42
				

*Please print clearly :*

Name :

Login :

@ucsc.edu

*No books ; No calculator ; No computer ; No email ; No internet ; No notes ; No phone. Do your scratch work elsewhere and enter only your final answer into the spaces provided. Points will be deducted for messy answers. Unreadable answers will be presumed incorrect.*

1. What are the four very general things a function might do when called ? [2✓]


2. *Ocaml*. Fill in the blanks : [2✓]

```
# List.fold_left;;
- : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>
```

```
# let sum = _____
val sum : int list -> int = <fun>
# sum [1;2;3;4;5];;
- : int = 15
```

```
# let length = _____
val length : 'a list -> int = <fun>
# length [1;2;3;4;5];;
- : int = 5
```

3. *Ocaml*. Define the function `fold_left`.

(a) *Ocaml*. [2✓]

(b) *Scheme*. [2✓]

4. Define the function `zipwith` that takes a function and two lists and uses the function to join the lists. If the lists are of different lengths, ignore the excess elements of the longer list. [2✓]

```
# zipwith;;
- : ('a -> 'b -> 'c) -> 'a list -> 'b list -> 'c list = <fun>
# zipwith (+) [1;2;3] [4;5;6;7];;
- : int list = [5; 7; 9]
```

5. **Ocaml.** Define the function **filter** which takes a predicate and a list and returns a list in the same order as the original list, but with only those elements for which the predicate is true. [2✓]

```
# filter;;
- : ('a -> bool) -> 'a list -> 'a list = <fun>
# filter ((>)0) [1;2;-3;-4;8;-3];;
- : int list = [-3; -4; -3]
```

6. **Ocaml.** Rewrite the second filter statement above using a **fun** (i.e., lambda) expression instead of `((>)0)`. [1✓]

7. **Smalltalk.** List the kinds of messages that can be sent to a Smalltalk object, and give an example of each. [3✓]

(a) Highest precedence :

(b) Middle precedence :

(c) Lowest precedence :

8. Fill in the following table. Some possible answers : John Backus. Alonzo Church. Grace Hopper. John Keme-ny & Thomas Kurtz. John McCarthy. Dennis Ritchie. Bjarne Stroustrup. [1✓]

Fortran	Cobol	Basic	Lisp

9. **Scheme.** Define the function **sum**. You may use the function **foldl** or explicitly write a tail-recursive function. [1✓]

10. **Java.** Give an example of how memory leak might happen in Java. [2✓]

11. **Ocaml.** Define the function **max** which returns **Some** maximum element of a list as determined by its function argument. Return **None** if the list is empty. **[3✓]**

```
# type 'a opt = None | Some of 'a;;
type 'a opt = None | Some of 'a
# max;;
- : ('a -> 'a -> bool) -> 'a list -> 'a opt = <fun>
# max (>) [3;1;4;1;5;9];;
- : int opt = Some 9
# max (<) [3;1;4;1;5;9];;
- : int opt = Some 1
# max (>) [];;
- : 'a opt = None
```

12. **Scheme.** Write a tail-recursive function called **reverse** that reverses a list. Do not use a higher-order function. **[2✓]**

```
> (reverse '(1 2 3 4 5))
(5 4 3 2 1)
```

13. **Scheme.** Define the function **map2** which takes three arguments: a binary function and two lists. It returns a list with the two lists merged using the function. If the lists are of different lengths, excess elements of the long list are ignored. **[2✓]**

```
> (map2 + '(1 2 3) '(4 5 6 7))
(5 7 9)
> (map2 * '(1 2 3 4 5) '(6 7 8))
(6 14 24)
```

14. **Ocaml.** Define **mul'** as per the project specifications. **[3✓]**

```
# add' [1;2;3] [4;5;6;7] 0;;
- : int list = [5; 7; 9; 7]
# add' [4;5;6;7] [2] 0;;
- : int list = [6; 5; 6; 7]
# add' [9;9;9;9] [1] 0;;
- : int list = [0; 0; 0; 0; 1]
```

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. **[12✓]**

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 $c = \max(a - b, 0)$	12		$= c$

- Replacing  $(\lambda x . + x 1) 2$  by  $(+ 2 1)$  is an example of
  - $\alpha$ -reduction
  - $\beta$ -reduction
  - $\eta$ -reduction
  - $\lambda$ -reduction
- The  $\lambda$ -calculus was invented by :
  - Ada Lovelace
  - Alan Turing
  - Alonzo Church
  - Grace Hopper
- A garbage collector automatically recycles what kind of objects ?
  - dead
  - live
  - reachable
  - unreachable
- What kind of polymorphism is `vector<string>` an example of ?
  - conversion
  - inclusion
  - overloading
  - parametric
- Two kinds of universal polymorphism are :
  - conversion & overloading
  - inclusion & parametric
  - overloading & inclusion
  - parametric & overloading
- Which of the following functions can be written tail-recursively without the need for reversing the list ?
  - filter
  - fold\_left
  - fold\_right
  - map
- Edsger Dijkstra published a paper entitled “\_\_\_\_\_ statement considered harmful”.
  - continue
  - goto
  - switch
  - throw
- Which was a language designed in the 1950s for use mainly in business data processing ?
  - 1957 FORTRAN
  - 1958 Algol 58
  - 1958 Lisp
  - 1959 COBOL
- The address of a static variable in C and C++ is determined when ?
  - compile time
  - link time
  - exec time
  - when a function is called
- Java generics and C++ templates are an example of \_\_\_\_\_ polymorphism.
  - conversion
  - inclusion
  - overloading
  - parametric
- Which C operator uses normal order evaluation ?
  - `++`
  - `<<`
  - `==`
  - `||`
- How will Ocaml respond to
 

```
# (+);;
```

  - `- : int * int * int = <fun>`
  - `- : int * int -> int = <fun>`
  - `- : int -> int * int = <fun>`
  - `- : int -> int -> int = <fun>`

