```
$Id: 2016q4-final,v 1.2 2016-12-06 12:03:00-08 - - $
Answers to cmps112-2016q4-final, page 1

________________________________________________________________________
Question 1. [3]

C++       |Bjarne Stroustrup's noted contribution to language design.
COBOL     |Business data processing language, designers Grace Hopper.
Algol 60  |Designed in Europe to express algorithms in a structured way.
C         |First version of Unix was 9000 lines of this language.
Lisp      |List processing language used in artificial intelligence.
FORTRAN   |Numeric and scientific computation language developed at IBM.
Simula 67 |Simulation language that influenced the design of C++.
Pascal    |Small language for structured programming by Niklaus Wirth.
Java      |Sun Micro claimed this language write once, run anywhere.

________________________________________________________________________
Question 2. [2]

arrow(a,b).
arrow(a,c).
arrow(b,c).
arrow(b,e).
arrow(c,d).
arrow(d,e).

________________________________________________________________________
Question 3. [2]

ispath(X,Y) :- arrow(X,Y).
ispath(X,Y) :- arrow(X,Z), ispath(Z,Y).

________________________________________________________________________
Question 4. [3]

findpath(X,Y,P) :- arrow(X,Y), P=[X,Y].
findpath(X,Y,P) :- arrow(X,Z), findpath(Z,Y,Q), P=[X|Q].

alternate:

findpath(X,Y,[X,Y]) :- arrow(X,Y).
findpath(X,Y,[X|Q]) :- arrow(X,Z), findpath(Z,Y,Q).

```

Answers to cmps112-2016q4-final, page 2

Question 5. [4]

```
let max gt list = match list with
    | [] -> failwith "max"
    | x::xs -> let rec max' x xs = match xs with
                    | [] -> x
                    | y::ys -> if gt x y then max' x ys
                                         else max' y ys
               in max' x xs
;;
```

Question 6. [2]

```
let rec zip x y = match x, y with
    | [], _ -> []
    | _, [] -> []
    | x::xs, y::ys -> (x,y)::zip xs ys
```

Question 7. [2]

```
let rec unzip list = match list with
    | [] -> ([],[])
    | (a,b)::rest -> let (al,bl) = unzip rest
                     in (a::al, b::bl);;
```

Question 8. [2]

```
gcd( X, Y, Z ) :- X > Y, T is X - Y, gcd( T, Y, Z ).
gcd( X, Y, Z ) :- X < Y, T is Y - X, gcd( X, T, Z ).
gcd( X, X, X ).
```

```

Answers to cmps112-2016q4-final, page 3

________________________________________________________________________
Question 9. [2]

universal    parametric (template, generic)
             inclusion (inheritance, OO)

ad hoc       conversion (coercion)
             overloading

________________________________________________________________________
Question 10. [2]

(define (pairthem l1 l2)
    (if (or (null? l1) (null? l2)) '()
        (cons (list (car l1) (car l2))
              (pairthem (cdr l1) (cdr l2)))))

________________________________________________________________________
Question 11. [6]

Object subclass: Stack [
   |array top|
   Stack class >> new [
      ^ Stack new: 10
   ]
   Stack class >> new: size [
      ^ super new init: size
   ]
   init: size [
      top := 0.
      array := Array new: size.
   ]
   pop [
      |result|
      result := array at: top.
      top := top - 1.
      ^ result.
   ]
   push: item [
      top := top + 1.
      array at: top put: item
   ]
   empty [
      ^ top = 0.
   ]
]

```

```

Answers to cmps112-2016q4-final, page 4

 1.    (A) APL

 2.    (B) Perl

 3.    (A) #!

 4.    (D) Lisp

 5.    (D) Simula 67

 6.    (B) 2.0 sqrt

 7.    (C) thunk

 8.    (B) 1958, John McCarthy.

 9.    (B) Ocaml

10.    (B) X = 1.2246467991473532e-16

11.    (B) duck-typing

12.    (D) virtual function table

```

```

Answers to cmps112-2016q4-final, page 4

 1.    (C) A structure on the heap, used to hold variables of an outer
           function when referenced by an inner function.

 2.    (B) Edsger Dijkstra

 3.    (D) \w+

 4.    (D) ? :

 5.    (A) (apply + '(1 2 3))

 6.    (A) function call stack

 7.    (D) throw

 8.    (D) reference counting

 9.    (B) race condition

10.    (D) val f : int -> int -> int -> int

11.    (A) (f '())

12.    (A) ('a -> 'b) -> 'a list -> 'b list

```