Name: _____ **Kerberos: _____**

```
;; Consider the following deriv function, that returns a new function that calculates
;; (an approximate) derivative of the function f:

(define (deriv f dx)
  (lambda (x) (/ (- (f (+ x dx))
                    (f (- x dx)))
                 (* 2 dx))))

(define (foo x) (* 3 x x x))
(define dfoo (deriv foo 0.001))

;; What will this print?
(print "first deriv of foo at 1:" (dfoo 1))



;; We'd like to generalize this to an nth-deriv function, so we can do things like:

(print "first deriv of foo at 1:" ((nth-deriv 1 foo 0.001) 1))

(print "fourth deriv of foo at 1:" ((nth-deriv 4 foo 0.001) 1))


;; Complete the implementation of nth-deriv:

(define (nth-deriv n f dx)









  )
```