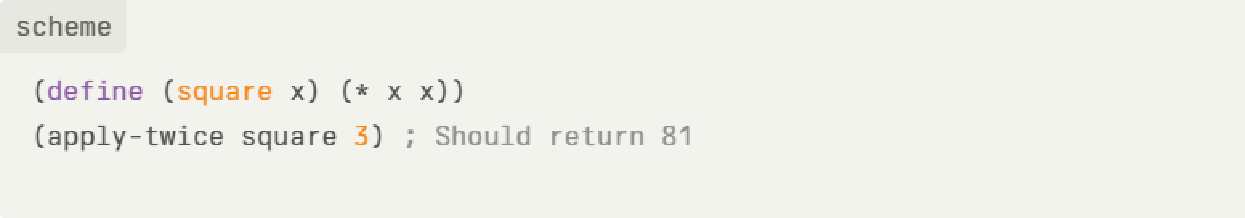
Programming Paradigms and Pragmatics (CS202)

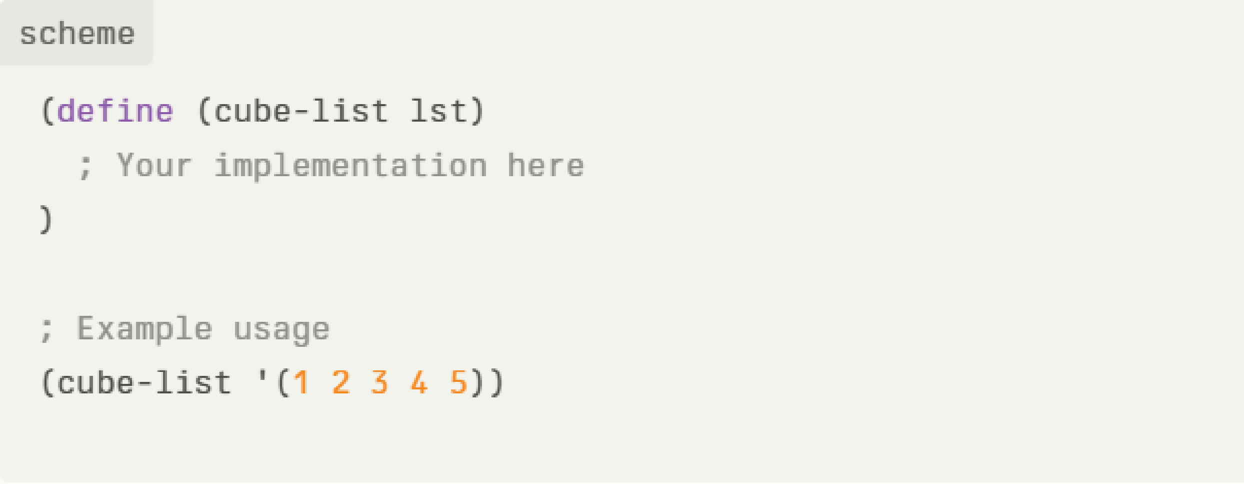
LAB - 5

1. Implement a tail-recursive version of the factorial function in Scheme. Your function should take a single parameter n and return n!.
2. Write a Scheme function called apply-twice that takes two parameters: a function f and a value x. The apply-twice function should apply the function f to x twice. For example:



```scheme
(define (square x) (* x x))
(apply-twice square 3) ; Should return 81
```

3. Implement a function in Scheme that uses the map higher-order function to cube all elements in a given list. Then, use this function to cube the elements of the list (1 2 3 4 5). Your implementation should include both the function definition and an example usage.



```scheme
(define (cube-list lst)
  ; Your implementation here
)

; Example usage
(cube-list '(1 2 3 4 5))
```