

Clay Ramey

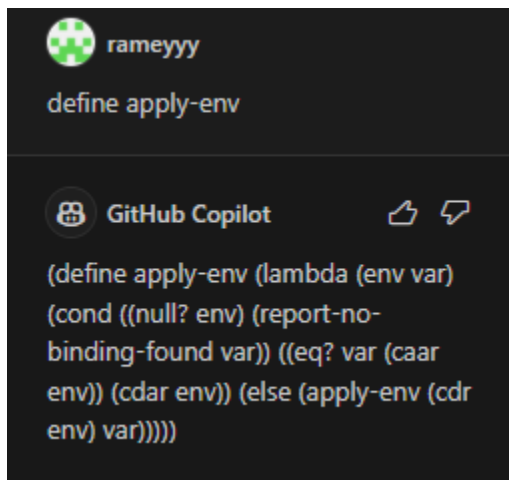
Dr. Sardinas

COMP-3220

HW6 | Copilot Bonus

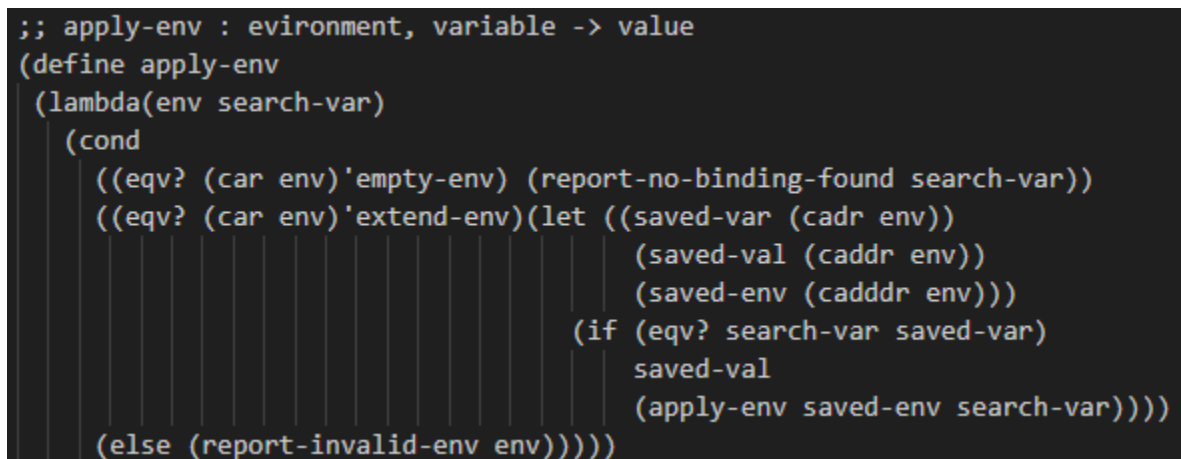
Using the help from Dr. Sardinas video tutorials, the empty-env and extend-env functions I created just defining a variable of type list for each in their respective functions, I didn't use copilot for these two functions.

The apply-env function was more difficult and after asking copilot, it spit out the following code:



```
(define apply-env (lambda (env var)
  (cond ((null? env) (report-no-binding-found var)) ((eq? var (caar env)) (cdar env)) (else (apply-env (cdr env) var)))))
```

I then looked up syntax for the scheme language and put this into proper syntax which gave me a working final deliverable, which looked like:



```
;; apply-env : environment, variable -> value
(define apply-env
  (lambda (env search-var)
    (cond
      ((eqv? (car env) 'empty-env) (report-no-binding-found search-var))
      ((eqv? (car env) 'extend-env) (let ((saved-var (cadr env))
                                          (saved-val (caddr env))
                                          (saved-env (cadddr env)))
                                     (if (eqv? search-var saved-var)
                                         saved-val
                                         (apply-env saved-env search-var))))
      (else (report-invalid-env env)))))
```