Homework 05

● Graded

Student
Giancarlos Marte

Total Points
100 / 100 pts

Autograder Score
85.0 / 85.0

Passed Tests
Exercise 1. s:subst applications (1) (4/4)
Exercise 1. s:subst applications (2) (4/4)
Exercise 1. s:subst lambda (1) (5/5)
Exercise 1. s:subst lambda (2) (4/4)
Exercise 1. s:subst numbers (5/5)
Exercise 1. s:subst variables (1) (4/4)
Exercise 1. s:subst variables (2) (4/4)
Exercise 2. s:eval apply (1) (5/5)
Exercise 2. s:eval apply (2) (10/10)
Exercise 2. s:eval lambda (5/5)
Exercise 2. s:eval lambda+apply (1) (5/5)
Exercise 2. s:eval numbers (5/5)
Exercise 3. e:eval apply (1) (3/3)
Exercise 3. e:eval apply (2) (12/12)
Exercise 3. e:eval lambdas (3/3)
Exercise 3. e:eval numbers (2/2)
Exercise 3. e:eval variables (5/5)

Question 2

Question 5 + 6 **15** / 15 pts

✔ **– 0 pts** Good

– 15 pts Missing

– 8 pts (5) Missing

Autograder Results

Exercise 1. s:subst applications (1) (4/4)

Exercise 1. s:subst applications (2) (4/4)

Exercise 1. s:subst lambda (1) (5/5)

Exercise 1. s:subst lambda (2) (4/4)

Exercise 1. s:subst numbers (5/5)

Exercise 1. s:subst variables (1) (4/4)

Exercise 1. s:subst variables (2) (4/4)

Exercise 2. s:eval apply (1) (5/5)

Exercise 2. s:eval apply (2) (10/10)

Exercise 2. s:eval lambda (5/5)

Exercise 2. s:eval lambda+apply (1) (5/5)

Exercise 2. s:eval numbers (5/5)

Exercise 3. e:eval apply (1) (3/3)

Exercise 3. e:eval apply (2) (12/12)

Exercise 3. e:eval lambdas (3/3)

Exercise 3. e:eval numbers (2/2)

Exercise 3. e:eval variables (5/5)

Submitted Files

hw5.rkt Download

```
#|
   ===> PLEASE DO NOT DISTRIBUTE THE SOLUTIONS PUBLICLY <===

  We ask that solutions be distributed only locally -- on paper, on a
  password-protected webpage, etc.

  Students are required to adhere to the University Policy on Academic
  Standards and Cheating, to the University Statement on Plagiarism and the
  Documentation of Written Work, and to the Code of Student Conduct as
  delineated in the catalog of Undergraduate Programs. The Code is available
  online at: http://www.umb.edu/life_on_campus/policies/code/

|#
;; PLEASE DO NOT CHANGE THE FOLLOWING LINES
#lang typed/racket
(provide (all-defined-out))
(require "hw5-util.rkt")
;; END OF REQUIRES

;; Exercise 1
(: s:subst (s:expression s:variable s:value -> s:expression))
(define (s:subst exp var val)
 (cond
   ; exp = num
   [(s:number? exp) exp]
   ; exp = var
   [(s:variable? exp)
    (cond
      [(equal? exp var) val]
      [else exp])]
   ; exp = (e1 e2)
   [(s:apply? exp)
    (define e1 (s:apply-func exp))
    (define e2 (s:apply-arg exp))
    (s:apply
     (s:subst e1 var val)
     (s:subst e2 var val))]
   ; exp = lambda
   [(s:lambda? exp)
    (define param (s:lambda-param exp))
    (define body (s:lambda-body exp))
    (cond
      [(equal? param var) exp]
      [else
       (s:lambda
        param
```

```
        (s:subst body var val))])]))

;; Exercise 2
(: s:eval ((s:expression s:variable s:value -> s:expression) s:expression -> s:value))
(define (s:eval subst exp)
  (cond
    ; exp = val
    [(s:value? exp) exp]
    ; exp = var
    [(s:variable? exp) (error "error")]
    ; exp = (ef, ea)
    [else
     (define ef (s:apply-func exp))
     (define ea (s:apply-arg exp))
     (define vf (s:eval subst ef))
     (define va (s:eval subst ea))
     (match vf
       [(s:lambda x ed)
        (s:eval subst (subst ed x va))])]))

;; Exercise 3
(: e:eval (e:environ e:expression -> e:value))
(define (e:eval env exp)
  (cond
    ; exp = val
    [(e:value? exp) exp]
    ; exp = var
    [(e:variable? exp) (e:env-get env exp)]
    ; exp = lambda
    [(e:lambda? exp)
     (define param (e:lambda-param exp))
     (define body (e:lambda-body exp))
     (e:closure env param body)]
    ; exp = (ef, ea)
    [else
     (define ef (e:apply-func exp))
     (define ea (e:apply-arg exp))
     (define vf (e:eval env ef))
     (define va (e:eval env ea))
     (match vf
       [(e:closure env2 param body)
        (e:eval
         (e:env-put env2 param va)
         body)])]))

;; Exercise 4 (Manually graded)
#|
Implementing lambda E is better for evaluating large and complex expressions. This is because
the run time of the search/find substitution in lambda S is linear. Meanwhile, the hash set
```

```
environment substitution of lambda E is constant time.

Implementing lambda S is better for really small and simple expression because the time and
space it would take for the search/find substitution would be smaller than the lambda E
hash set environment substitution.
|#

;; Exercise 5 (Manually graded)
#|
Using a formal specification helps with preventing and finding bugs before making code.
Another benefit is that it gives you a way to think about how to design a program which reduces
ambiguity and makes code more concise.
|#

```