CSE 413 HW 2

Yang Zhang

**Part I.**

**1**

\*\*\* pre definitions

\* (define x 5)

\* (define y 7)

\* (define z 11)

\*\*\*

(1) (let ((x 0) (z (\* x z))) (+ x y z))

process: local x is set to 0, local z is set to globe x times globe z

therefore (+ x y z) 🡪 (+ 0 7 55) 🡪 62

(2) (let ((x 2) (y (- x 4)) (z (\* y 2))) (+ x y z))

process: local (x y z) is set to (2 1 14)

therefore (+ x y z) 🡪 (+ 2 1 14) 🡪 17

(3) (let\* ((x 2) (y (- x 4)) (z (\* y 2))) (+ x y z))

process: globe (x y z) is set to (2 -2 -4)

therefore (+ x y z) 🡪 (+ 2 -2 -4) 🡪 -4

**2**

\*\*\* Pre-definition \*\*\*

(define make-thing   
      (lambda (thing f)  
      (lambda (x)  
         (\* thing (f x)))))  
  
(define double  
   (lambda (x) (+ x x)))

(define mystery (make-thing 3 double))

(a) (make-thing 3 double) works as curry function that is the function that take another function as an argument. As a result, this function would multiply whatever value passed in by 6

(b) (mystery 5) prints 30. make-thing will do (argument, function(x)) and would multiply argument with the value returned by function(x). In definition of ***mystery***, 3 as argument and ***double*** as function are passed in. When evaluate (mystery 5), 3 and ***double(5)*** are passed in ***make-thing*** , therefore (mystery 5) 🡪 (\* 3 (double 5)) 🡪 30