```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; Structure and Interpretation of Computer Programs, 2. ed.      ;;;;;
;;;;; Instructor Manual, Section 1.1, Exercise M1.1                  ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;; Student: Abrantes Araújo Silva Filho                           ;;;;;
;;;;; Date: 2019-02-11                                               ;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;;;; Evaluate the expressions. Give the result the interpreter would
;;;; print or else explain why the avaluation would cause an error.

(* (+ 2 2)       5)
; 20

(* (+ 2 2) (5))
; ERROR. The subexpression (5) is translated as "apply 5", but the
; object 5 is not applicable.

(*(+(2 2) 5))
; ERROR. The subexpression (2 2) is translate as "apply 2 to 2", but
; the (first) 2 is not applicable. NOTE: spaces are not always
; mandatory, for example: (*(+(+ 2 2)5)1) => 9

(*(+ 2
   2)5)
; 20

(5 * 4)
; ERROR. The object 5 is not applicable.

(5 * (2 + 2))
; ERROR. The object 2 is not applicable. NOTE: the interpreter first
; evaluate the subexpression 5, *, and (2 + 2) (before it tries to
; apply the first subexpression to the others). Then, when the
; interpreter evaluates (2 + 2), it gives the error.

((+ 2 3))
; ERROR. The object 5 is not applicable.
```