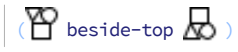
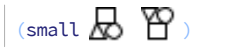
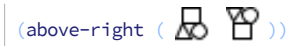
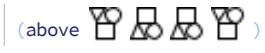
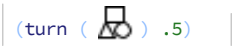
; ● CSC104 Winter 2020 Exercise #1 ●

; Print this out and fill it in by hand. Hand in your solutions to the TA at the start of your quiz.

; UTorID :
; Surname :
; Given Name :

; Precision and care are crucial in programming, and we assume you check your exercise answers in DrRacket.
; Your mark will reflect the care you took to make sure your answers are all, or almost all, correct.

; ● Part A. Circle each of the following twelve pieces of code that reports an error (rather than produces a value) ...

	`(+ 1 (2 - 3) 4)`		
`(* 1 2 (- 3) 4)`			
			

; ● Part B. Show all the steps to evaluate the following expression.

; You do not need to include the "● Steps ●", "○", nor "•" punctuation that DrRacket shows when using step.
; Include the underlining of sub-expressions that will change.
; In DrRacket, the step operation starts by copying the given expression so that it can add some underlining,
; but you may save some effort by adding the initial underlining directly to the original expression.

```
(mirror (beside (clockwise (tall (solid-triangle (/ 150 10))))
                (turn (above (circle (+ 10 40) "solid" "red")
                             (square (- 30 15)))
                      (+ (* 2 1 5) (* (height (wide (circle 10))) 3) 5))))
```

; ● Part C. Beside each of the following expressions, write its value ...

△

```
unary?
-123
circle
width
45
number?
*
#true
turn
(function? -67)
(image? square)
(boolean? ⊠)
(function? rectangle)
(number? height)
(number? +)
(function? /)
(image? image?)
(function? function?)
(boolean? image?)
(boolean? boolean?)
(image? ◆)
(boolean? #false)
(function? boolean?)
(number? -89)
(image? #true)
(unary? scale-height)
(binary? solid-oval)
(binary? -)
(unary? binary?)
```

; ● Part D. Show all the steps to evaluate the following expressions.
; Include the underlining of sub-expressions that will change.

; You do not need to include the "● Steps ●", "○", nor "•" punctuation that DrRacket shows when using step .

; In DrRacket, the step operation starts by copying the given expression so that it can add some underlining,
; but you may save some effort by adding the initial underlining directly to the original expression.

```
(image? (+ 1 2 3))
```

```
(number? (circle 10))
```

```
(boolean? (- 45))
```

```
(function? (flip △))
```



```
(image? (rectangle 20 10))
```

```
(number? (/ 10 2))
```

```
(boolean? (unary? beside-top))
```

```
(function? (image? 12))
```

```
(image? (image? mirror))
```

; ● Part E.
; For each definition, circle either “Function” or “Variable” according to whether it is a function definition or a variable definition.
; If it defines a variable, write down the variable name.
; If it defines a function, write down the function name and parameter names.

```
(define (s z y)
  (text-join z y))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define (b d c) (above d c (turn d 45)))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define f
  (text-join
   b
   "b"))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define i (square 10 "solid" "black"))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define x
  (b i j))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define
  rick
  "rick")

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define
  (%-width an-image %)
  (* (/ % 100)
     (width an-image)))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define sun (scale
             3))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define bottom (remove-bottom kids
                              (+ 1 (/
                                    (height kids)
                                    2))))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```



```
(define (x g)
  (turn (above x x)
        45))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define (raise
  (above i (flip (triangle (width i)))))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define another-good-number
  (* 2 52))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define (remove-bottom an-image
                       a-bottom)
  (image-top an-image (- (height an-image)
                         (height a-bottom))))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define (scaled-bird
         amount)
  (scale
   
   amount))

; Defines a ...  Function  Variable

; Variable or Function Name:

; Parameter Names (if any):
```

```
(define (b x y) (+ (text-length y)
                   x))

; Defines a:  Function  Variable

; Variable Name (if applicable):

; Variable or Function Name:

; Parameter Names (if any):
```