Tyler Cullen
CSC 344
2/18/2023

# Racket Programming Assignment

## Interactions, Definitions, Applications

---

## Abstract

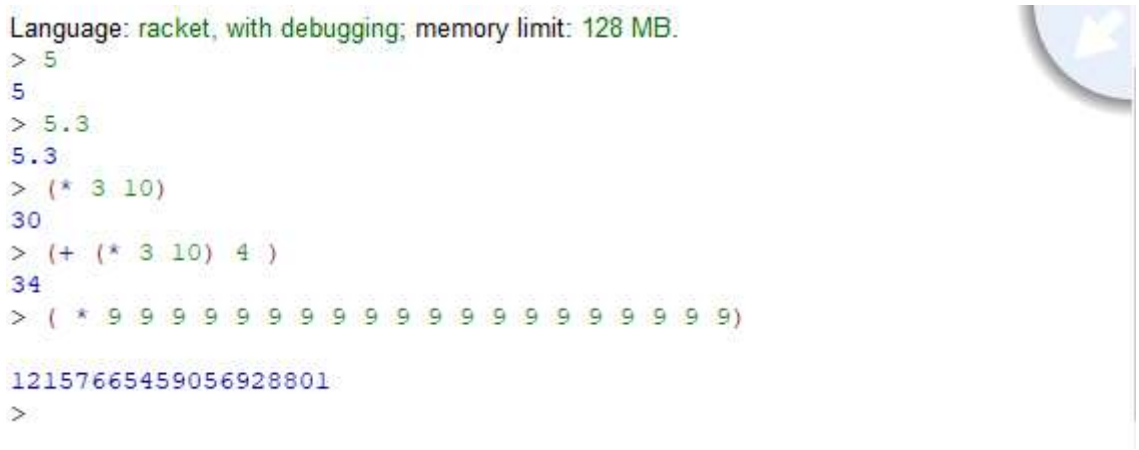
This assignment is about doing some simple Racket programming. It will involve various examples such as , writing a number of function definitions and engaging in computational problem solving. The programs will involve basic arithmetic and creating different colored shapes, and sizes of shapes.

---

## Task 1 - Scrap of Tin

---

## “Arithmetic Expressions”



## “Solve a Simple Problem”

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> pi
3.141592653589793
> side
side: undefined;
 cannot reference an identifier before its definition
> (define side 100)
> side
100
> (define square-area (* side side  ) )
> square-area
10000
> ( define radius ( / side 2 ) )
> radius
50
> ( define circle-area ( * pi radius radius ) )
> circle-area
7853.981633974483
> (define scrap-area ( - square-area circle-area ) )
> scrap-area
2146.018366025517
>
```

Determine language from source ▾

## “Rendering an Image of the Problem Situation”

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (require 2htdp/image)
> (define side 100)
> (define the-square (square side "solid"  "silver"))
> the-square
```

```
> (define radius (/ side 2))
> (define the-circle (circle radius "solid" "white"))
> (define the-image (overlay the-circle the-square))
> the-image
```

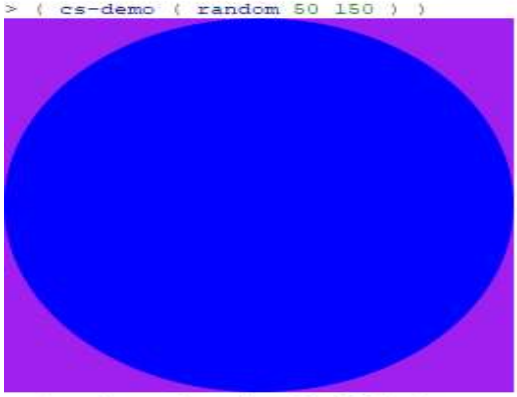
```
>
```

Determine language from source ▾

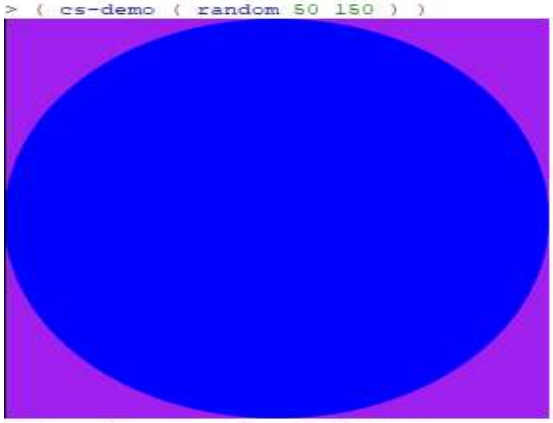
## Task 2: Definitions - Inscribing/Circumscribing Circles/Squares

### CS-Demo

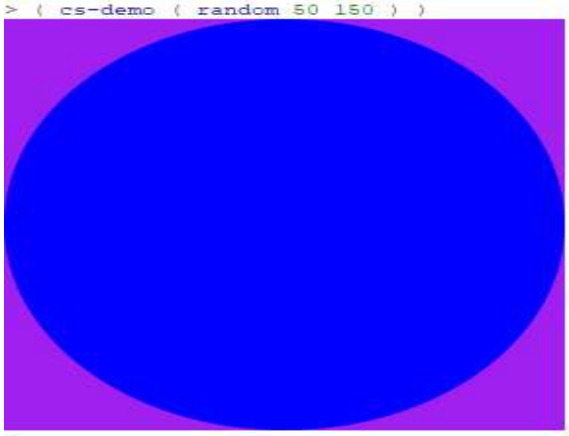
```
> ( cs-demo ( random 50 150 ) )
```



```
> ( cs-demo ( random 50 150 ) )
```



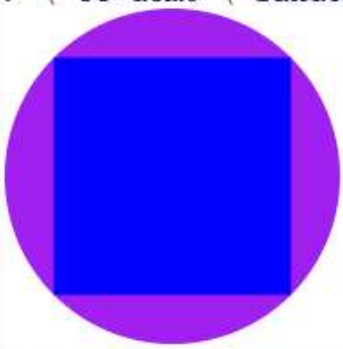
```
> ( cs-demo ( random 50 150 ) )
```



```
>
```

**CC-Demo**

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
#<procedure:>>
> ( cc-demo ( random 50 150 ) )
```



```
> ( cc-demo ( random 50 150 ) )
```

```
> ( cc-demo ( random 50 150 ) )
```

```
>
```

**IC DEMO**

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
#<procedure:>>
> ( ic-demo ( random 50 150 ))
```

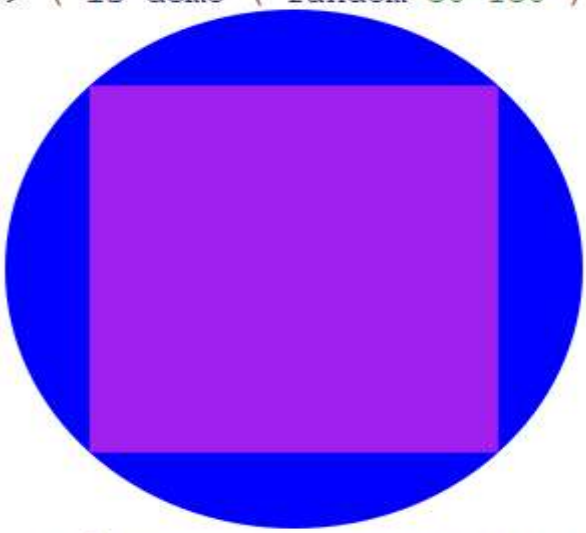
```
> (ic-demo ( random 50 150 ))
```

```
> (ic-demo ( random 50 150 ))
```

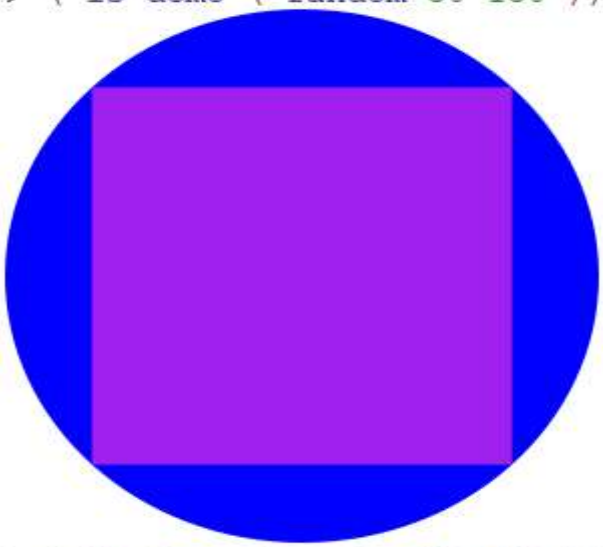
```
>
```

**IS DEMO**

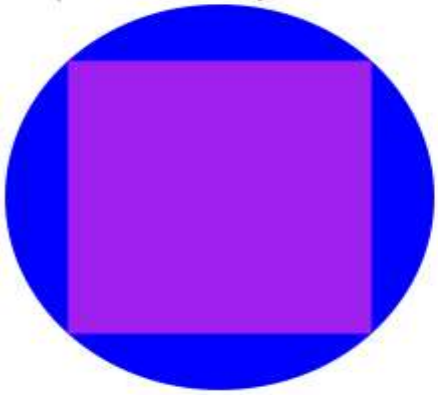
```
#<procedure:>>
> ( is-demo ( random 50 150 ))
```



```
> ( is-demo ( random 50 150 ))
```



```
> ( is-demo ( random 50 150 ))
```



```
>
```

## CODE

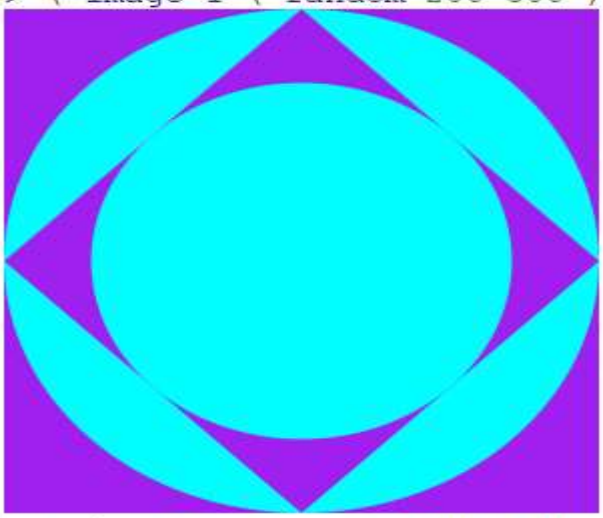
```
#lang racket
> ( require 2htdp/image )
( define ( cs radius ) ( * radius 2 ))
( define ( cc side ) ( / ( * side ( sqrt 2) )2 ) )
( define ( ic side ) ( / side 2.0 ) )
( define ( is radius ) (/ ( * radius 2 ) ( sqrt 2 )))

( define ( cs-demo radius ) ( overlay ( circle radius "solid" "blue") (square ( cs radius ) "solid" "purple" ) ) )
( define ( cc-demo side ) ( overlay ( square side "solid" "blue" ) ( circle ( cc side ) "solid" "purple" ) ) )
( define ( ic-demo side ) ( overlay ( circle ( ic side ) "solid" "purple" ) ( square side "solid" "blue" ) ) )
( define (is-demo radius) ( overlay ( square (is radius) "solid" "purple") ( circle radius "solid" "Blue" ) ) )
```

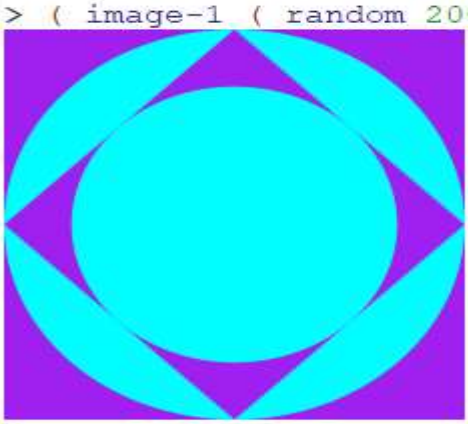
## Task 3: Definitions - Inscribing/Circumscribing Images

---

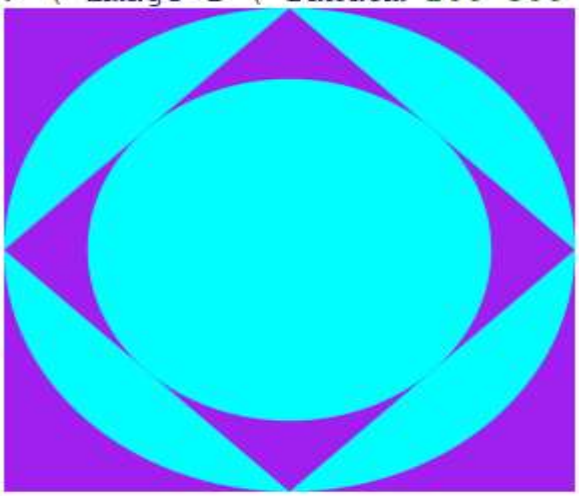
```
> ( image-1 ( random 200 300 ))
```



```
> ( image-1 ( random 200 300 ))
```



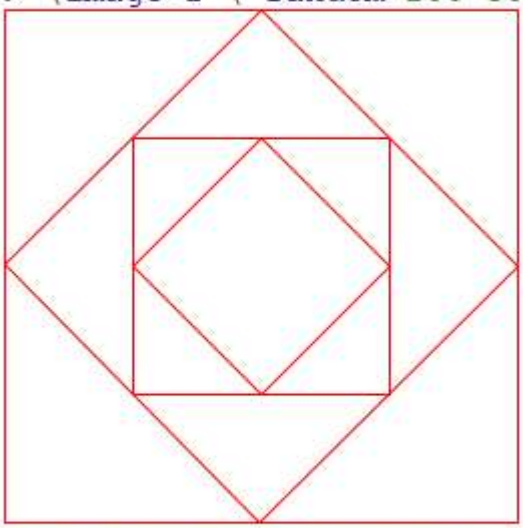
```
> ( image-1 ( random 200 300 ))
```



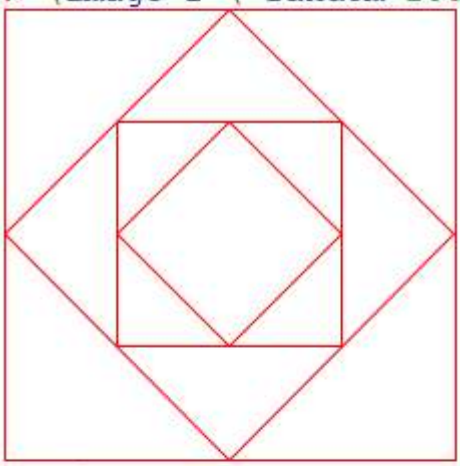
```
>
```

Language: racket, with debugging; memory limit: 128 MB.

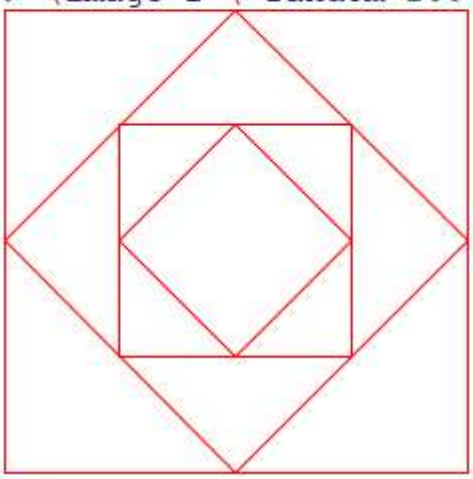
```
#<procedure:>>
> (image-2 ( random 200 300 ))
```



```
> (image-2 ( random 200 300 ))
```

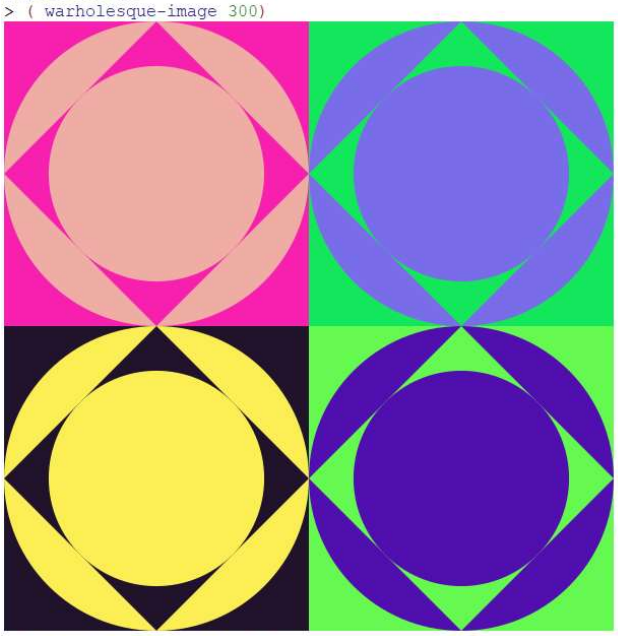


```
> (image-2 ( random 200 300 ))
```

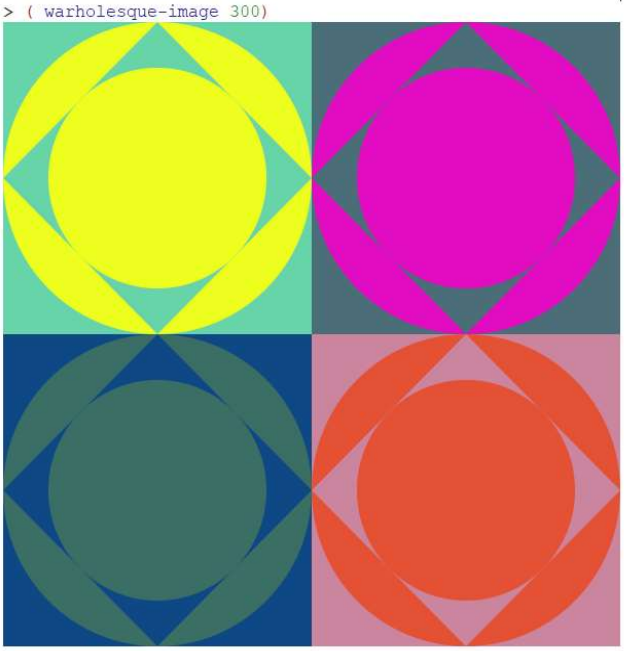


```
> |
```

Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
#<procedure:>>
> ( warholesque-image 300)



> ( warholesque-image 300)



>

# CODE

```
#lang racket
> ( require 2htdp/image )
( define ( cs radius ) ( * radius 2 ))
( define ( cc side ) ( / ( * side ( sqrt 2) )2 ) )
( define ( ic side ) ( / side 2.0 ) )
( define ( is radius ) (/ ( * radius 2 ) ( sqrt 2 )))

( define ( cs-demo radius ) ( overlay ( circle radius "solid" "blue") (square ( cs radius ) "solid" "purple" ) ) )
( define ( cc-demo side ) ( overlay ( square side "solid" "blue" ) ( circle ( cc side ) "solid" "purple" ) ) )
( define ( ic-demo side ) ( overlay ( circle ( ic side ) "solid" "purple" ) ( square side "solid" "blue" ) ) )
( define (is-demo radius) ( overlay ( square (is radius) "solid" "purple") ( circle radius "solid" "Blue" ) ) )

( define ( image-1 side )
   ( overlay ( circle ( ic ( is ( ic side ) ) ) "solid" "cyan" )
             (overlay( rotate 45 ( square ( is ( ic side ) ) "solid" "purple" ) )
                      ( overlay ( circle ( ic side ) "solid" "cyan" ) ( square side "solid" "purple" ) ) ) ) )

( define ( image-2 side )
   ( define box-2 ( is ( ic side ) ) )
   ( define box-3 ( is ( ic box-2 ) ) )
   ( define box-4 ( is ( ic box-3 ) ) )
   ( overlay ( rotate 45 ( square box-4 "outline" "red"  ) )
             ( overlay ( square box-3 "outline" "red" )
                       ( overlay( rotate 45 ( square box-2 "outline" "red" ) )
                                ( overlay ( square side "outline" "red" )
                                          ( square side "outline" "red") ) ) ) ) )
( define ( single-image side )
   ( define ( random-color ) ( color ( random 256 ) ( random 256 ) ( random 256 ) ) )
   ( define color-1 ( random-color ) )
   ( define color-2 ( random-color ) )
   ( overlay ( circle ( ic ( is ( ic side ) ) ) "solid" color-2 )
             ( overlay ( rotate 45 ( square ( is ( ic side ) ) "solid" color-1 ) )
                       ( overlay ( circle ( ic side ) "solid" color-2 ) ( square side "solid" color-1 ) ) ) ) )


( define ( warholesque-image side )
   ( above (beside (single-image side ) ( single-image side ) )
           (beside(single-image side ) ( single-image side ) ) ) )
```

## Task 4: Permutations of Randomly Colored Stacked Dots

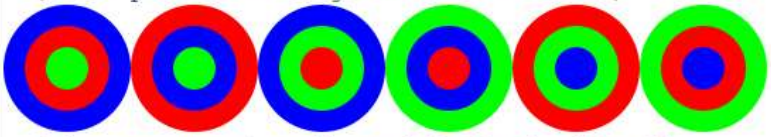
```
Language: racket, with debugging; memory limit: 128 MB.
#<procedure:>>
> ( tile "grey" "red" "green" "blue" )
```

```
> ( tile "red" "green" "yellow" "white" )
```

```
> ( tile "black" "orange" "purple" "grey" )
```

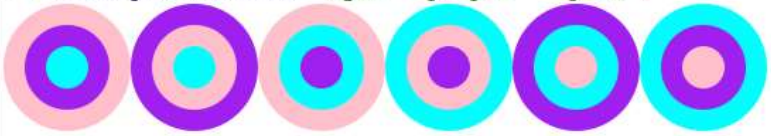
```
>
```

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
#<procedure:>>
> ( dots-permutations "green" "red"  "blue")
```



```
> ( dots-permutations "brown" "yellow"  "black")
```

```
> ( dots-permutations "cyan" "purple"  "pink")
```



```
>
```

# CODE

Untitled ▾ (define ...) ▾

```
1  #lang racket
2  > ( require 2htdp/image )
3
4  (define (tile color1 color2 color3 color4)
5  (overlay (circle 15 "solid" color4)
6  (overlay (circle 30 "solid" color3)
7  (overlay ( circle 45 "solid" color2)
8  (square 100 "solid" color1) ) )  ))
9
10 (define (dot-permutation color1 color2 color3)
11 (overlay (circle 15 "solid" color1)
12 (overlay (circle 30 "solid" color2)
13 (circle 45 "solid" color3) ) ) )
14
15 (define (dots-permutations color1 color2 color3)
16 (beside (dot-permutation color1 color2 color3)
17 (beside (dot-permutation color1 color3 color2)
18 (beside (dot-permutation color2 color1 color3)
19 (beside (dot-permutation color2 color3 color1)
20 (beside (dot-permutation color3 color1 color2)
21         (dot-permutation color3 color2 color1) ) ) ) ) ))
```