Declan ONUNKWO 02/21/2023

CSC344 – Programming Languages

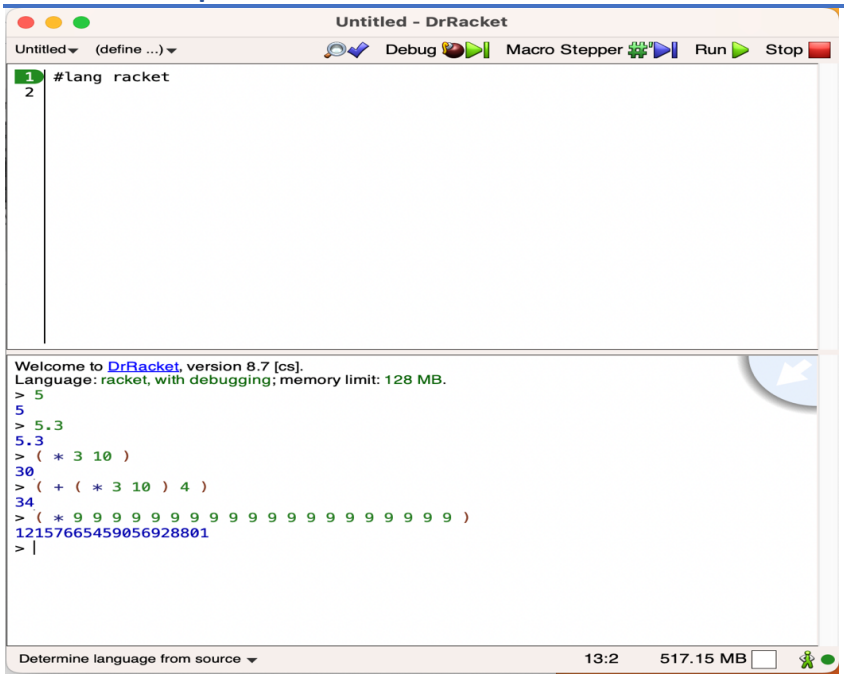
Assignment: **Racket Assignment 2 – Interactions, Definitions, Applications**

LEARNING ABSTRACT

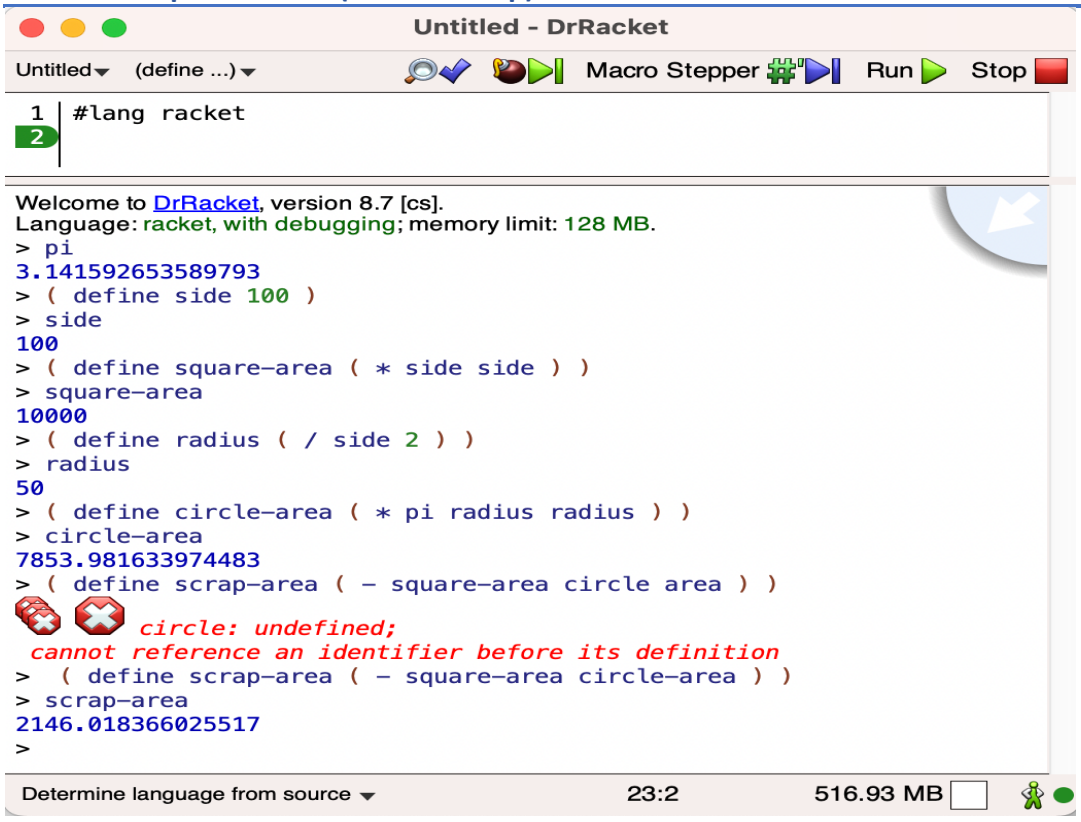
The aim of this assignment is to improve one's understanding of the Racket program and its syntax. This assignment will cover the interactions, definitions and applications in a racket programming language. The first task would be a simple copy and follow, while the remaining three tasks would engage the student more with the programming language by testing their ability to solve problems and follow instructions while working with Racket.

Task 1: Interactions - Scrap of Tin

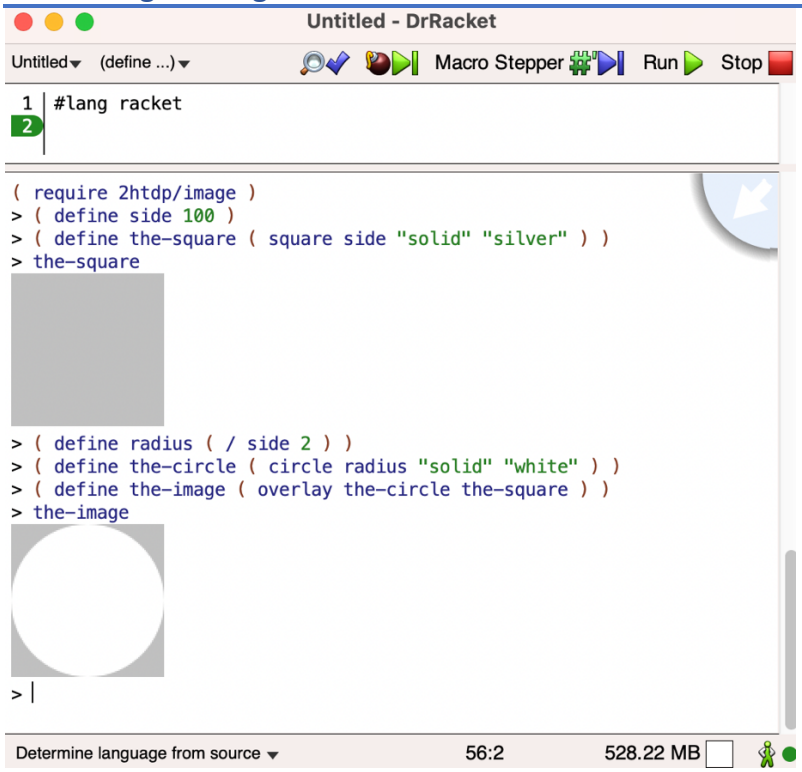
Arithmetic Expressions



Solve a Simple Problem (Area of Scrap)

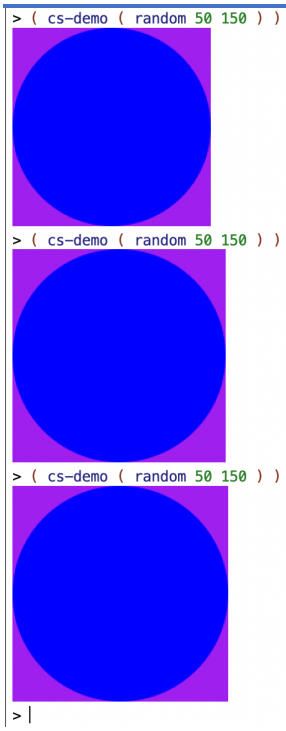


Rendering an Image of the Problem Situation



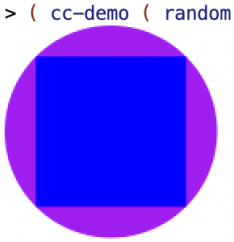
Task 2: Definitions - Inscribing/Circumscribing Circles/Squares

cs-demo

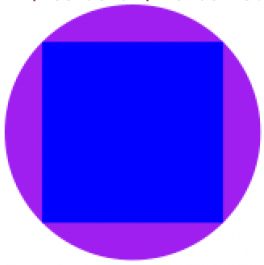


cc-demo

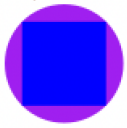
```
> ( cc-demo ( random 50 150 ) )
```



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



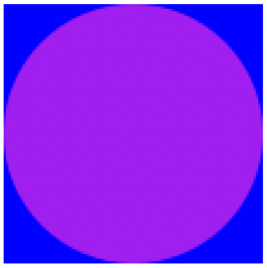
```
> ( cc-demo ( random 50 150 ) )
```



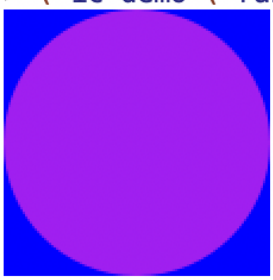
```
>
```

ic-demo

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



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

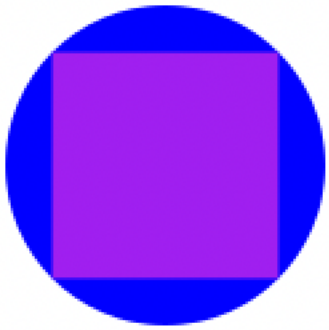


```
> ( ic-demo ( random 50 150 ) )
```

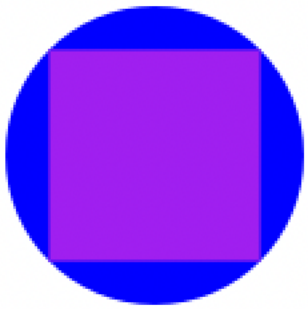
```
>
```

is-demo

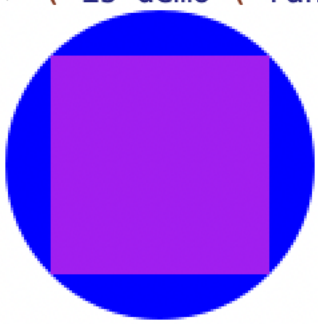
```
> ( is-demo ( random 50 150 ) )
```



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



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



```
>
```

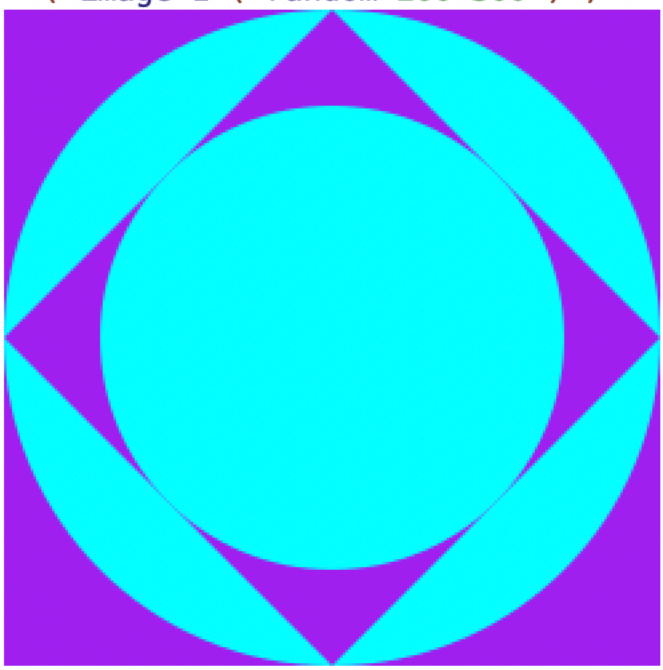
The Code

```
#lang racket
( require 2htdp/image )

( define ( cs radius )
    ( * radius 2 )
)

( define ( cc side-length )
  ( sqrt ( * ( / side-length 2 ) ( / side-length 2 ) 2 ) )
)

( define ( ic square-side )
  ( / square-side 2 )
)

( define ( is cir-radius )
  ( / ( * cir-radius 2 ) ( sqrt 2 ) )
)

( define ( cs-demo csDemoRadius )
  ( define cir ( circle csDemoRadius "solid" "blue" ) )
  ( define sqr ( square ( cs csDemoRadius ) "solid" "purple" ) )
  ( overlay cir sqr )
)

( define ( cc-demo ccDemoSide )
  ( define sqr ( square ccDemoSide "solid" "blue" ) )
  ( define cir ( circle ( cc ccDemoSide ) "solid" "purple" ) )
  ( overlay sqr cir )
)

( define ( ic-demo icDemoSide )
  ( define cir ( circle ( ic icDemoSide ) "solid" "purple" ) )
  ( define sqr ( square icDemoSide "solid" "blue" ) )
  ( overlay cir sqr )
)

( define ( is-demo isDemoRadius )
  ( define sqr ( square ( is isDemoRadius ) "solid" "purple" ) )
  ( define cir ( circle isDemoRadius "solid" "blue" ) )
  ( overlay sqr cir )
)
```

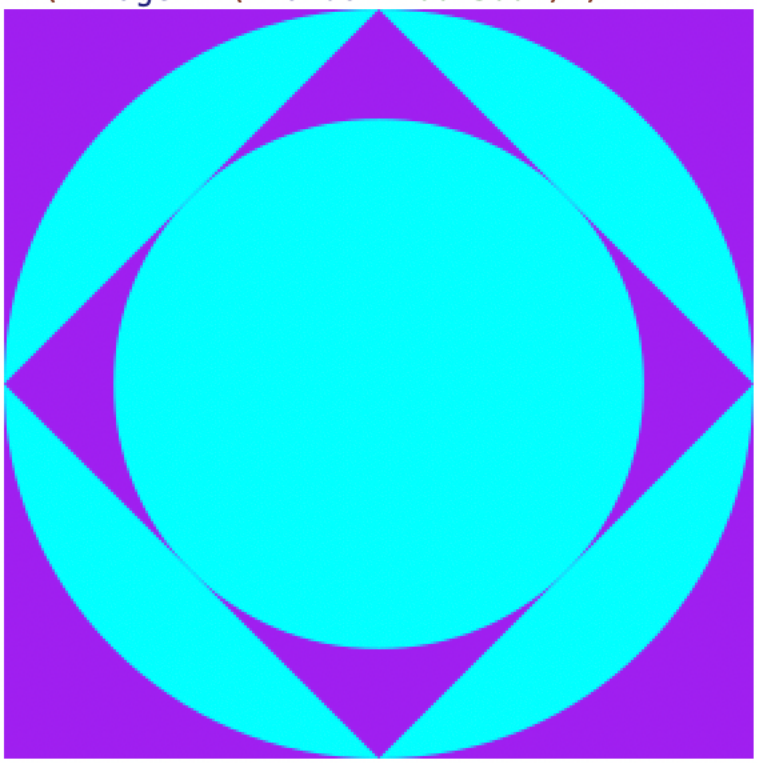
Task 3: Inscribing/Circumscribing Images

Image 1 Demo

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



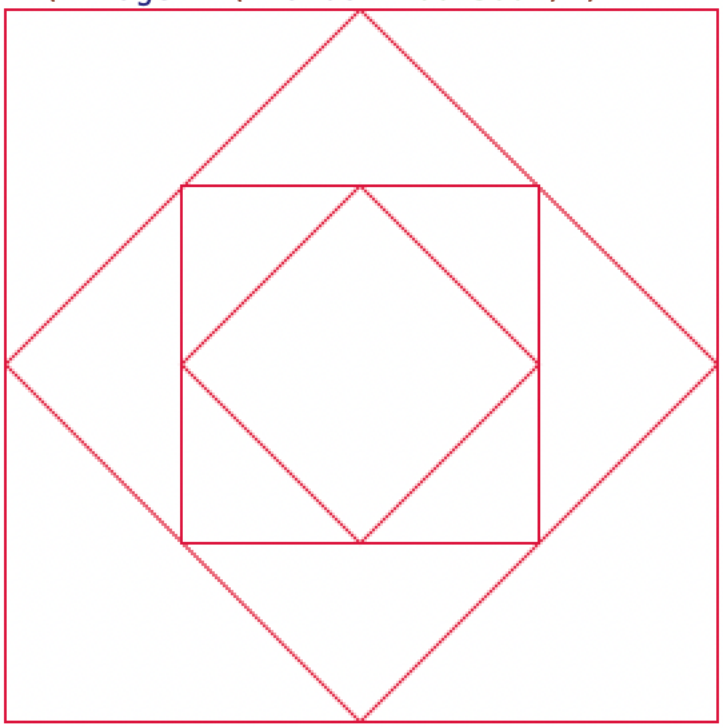
```
> ( image-1 ( random 200 300 ) )
```



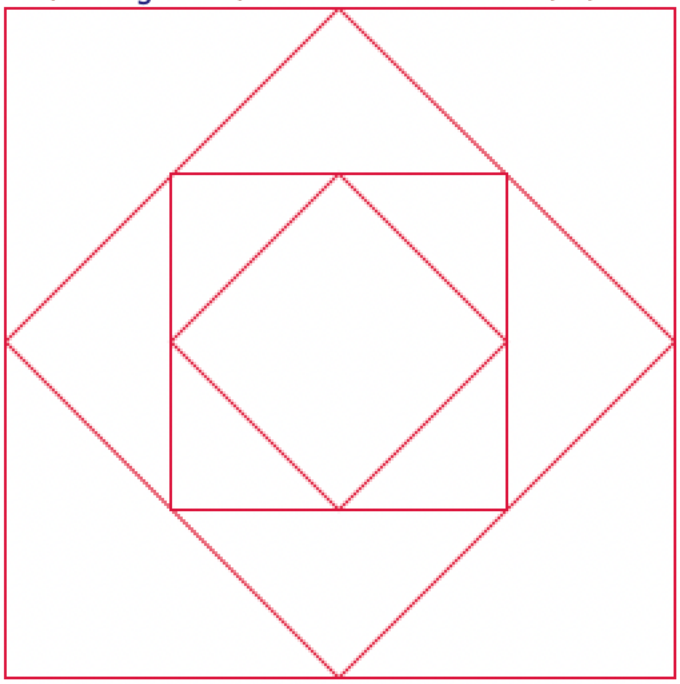
```
> 
```

Image 2 Demo

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



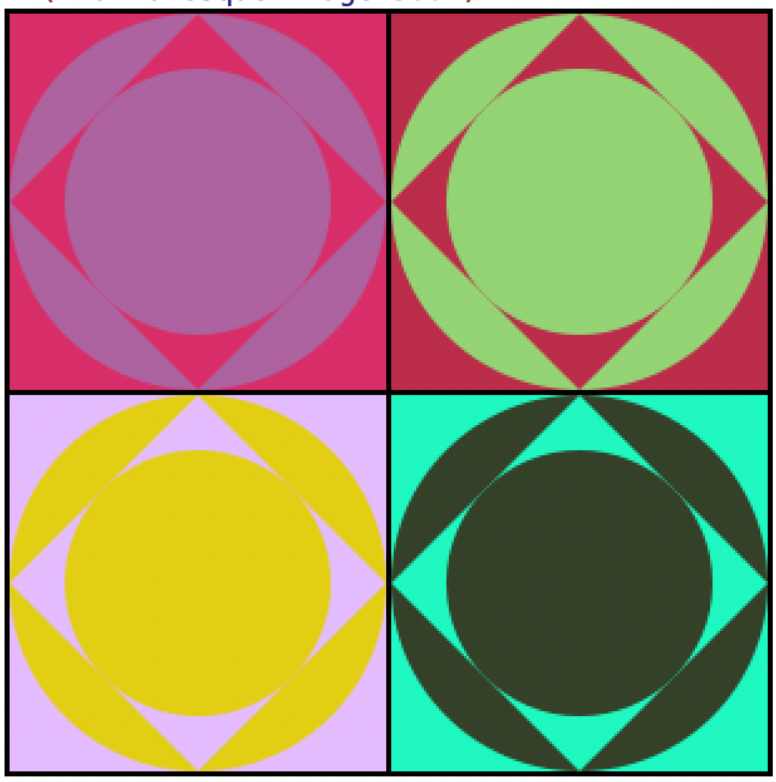
```
> ( image-2 ( random 200 300 ) )
```



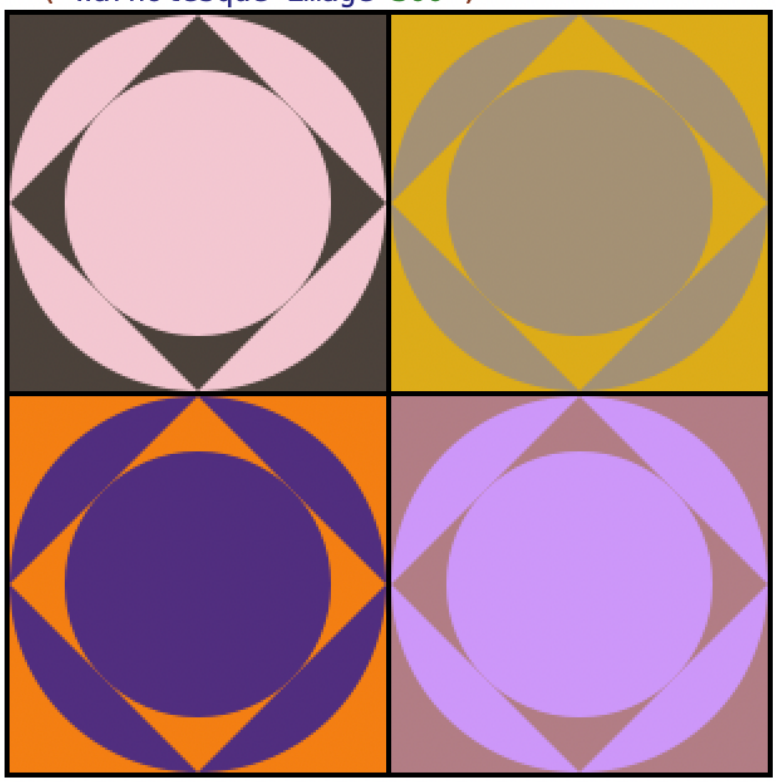
```
>
```

Warholesque Image

```
> ( Warholesque-image 300 )
```



```
> ( Warholesque-image 300 )
```



```
>
```

The Code

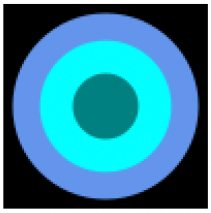
```
( define ( image-1 sqrSide )
  ( define cir1 ( circle ( ic sqrSide ) "solid" "cyan" ) )
  ( define sqr1 ( square sqrSide "solid" "purple" ) )
  ( define sqr2 ( rotate 45 ( square ( is ( ic sqrSide ) ) "solid" "purple" ) ) )
  ( define cir2 ( circle ( ic ( is ( ic sqrSide ) ) ) "solid" "cyan" ) )
  ( overlay cir2 sqr2 cir1 sqr1 )
)

( define ( image-2 sqrSide )
  ( define sqr1 ( square sqrSide "outline" "crimson" ) )
  ( define sqr2 ( rotate 45 ( square ( is ( ic sqrSide ) ) "outline" "crimson" ) ) )
  ( define sqr3 ( square ( is ( ic ( is ( ic sqrSide ) ) ) ) "outline" "crimson" ) )
  ( define sqr3-side (  is ( ic ( is ( ic sqrSide ) ) ) ) )
  ( define sqr4 ( rotate 45 ( square ( is ( ic sqr3-side ) ) "outline" "crimson" ) ) )
  ( overlay sqr4 sqr3 sqr2 sqr1 )
)

( define ( Warholesque-image canvasSide )
  ( define ( image1 canvasSide )
  ( define sqrSide ( / canvasSide 2 ) )
  ( define (random-color) ( color ( random 0 256 ) ( random 0 256 ) ( random 0 256 ) ) )
  ( define circleColor ( random-color ) )
  ( define squareColor ( random-color ) )
  ( define border ( square ( + 2 sqrSide ) "solid" "black" ) )
  ( define cir1 ( circle ( ic sqrSide ) "solid" circleColor ) )
  ( define sqr1 ( square sqrSide "solid" squareColor ) )
  ( define sqr2 ( rotate 45 ( square ( is ( ic sqrSide ) ) "solid" squareColor ) ) )
  ( define cir2 ( circle ( ic ( is ( ic sqrSide ) ) ) "solid" circleColor ) )
  ( overlay cir2 sqr2 cir1 sqr1 border )
    )
  ( define border1 ( square ( + 6 canvasSide ) "solid" "black" ) )
  ( overlay
  ( above
   ( beside ( image1 canvasSide ) ( image1 canvasSide ) )
   ( beside ( image1 canvasSide ) ( image1 canvasSide ) )
   )
  border1
  )
)
```

Task 4: Permutations of Randomly Colored Stacked Dots

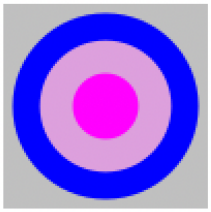
Demo

Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.

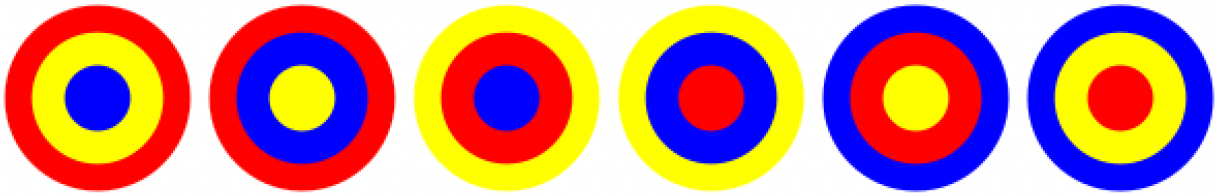
```
> ( tile "black" "cornflowerblue" "cyan" "teal" )
```



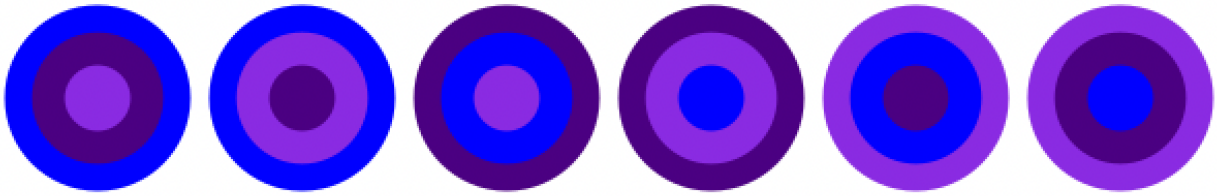
```
> ( tile "silver" "blue" "plum" "magenta" )
```



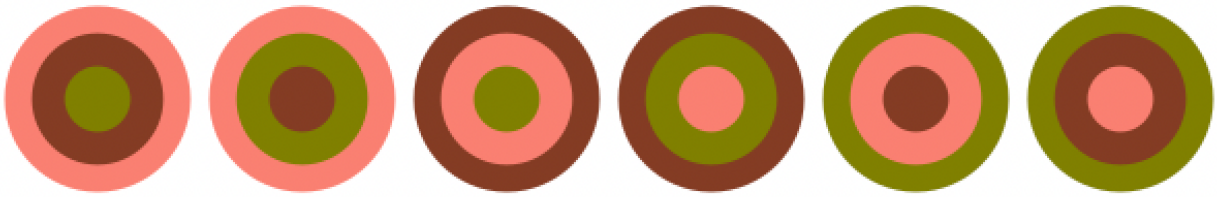
```
> ( dot-permutations "red" "yellow" "blue" )
```



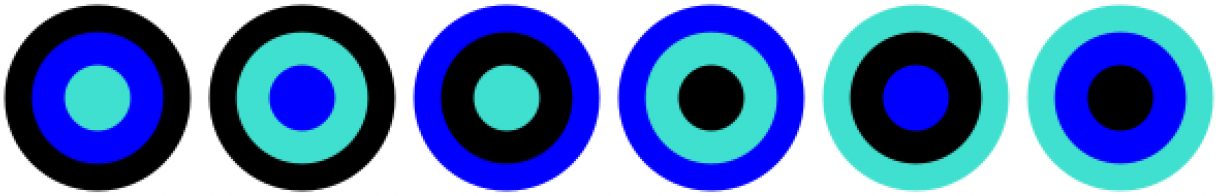
```
>  ( dot-permutations "blue" "indigo" "blueviolet" )
```



```
>  ( dot-permutations "salmon" "brown" "olive" )
```



```
>  ( dot-permutations "black" "blue" "turquoise" )
```



```
>
```

Code

TilesAndCircles.rkt (define ...)

```
#lang racket
( require 2htdp/image )

( define ( ic side-length )
   ( / side-length 2.0 )
)

( define ( cir2 radius )
   ( / ( / ( * radius 2 ) ( sqrt 2 ) ) 2 )
)

( define ( cir3 radius )
   ( / ( / ( * ( cir2 ( cir2 radius ) ) 2 ) ( sqrt 2 ) ) 2 )
)

( define ( tile sqrColor cir1Color cir2Color cir3Color )
   ( define radius 45 )
   ( define sqr ( square ( * radius 2.2 ) "solid" sqrColor ) )
   ( define c1 ( circle radius "solid" cir1Color ) )
   ( define c2 ( circle ( cir2 radius ) "solid" cir2Color ) )
   ( define c3 ( circle ( cir3 radius ) "solid" cir3Color ) )
   ( overlay c3 c2 c1 sqr )
)

( define ( mixer color1 color2 color3 )
      ( define radius 45 )
      ( define sqr ( square ( * radius 2.2 ) 0 color1 ) )
      ( define c1 ( circle radius "solid" color1 ) )
      ( define c2 ( circle ( cir2 radius ) "solid" color2 ) )
      ( define c3 ( circle ( cir3 radius ) "solid" color3 ) )
      ( overlay c3 c2 c1 sqr )
      )

( define ( dot-permutations color1 color2 color3 )
   ( beside ( mixer color1 color2 color3 ) ( mixer color1 color3 color2 )
            ( mixer color2 color1 color3 ) ( mixer color2 color3 color1 )
            ( mixer color3 color1 color2 ) ( mixer color3 color2 color1 )
            )
)
```