Temitope Emokpae 2/21/2023

CSC 344

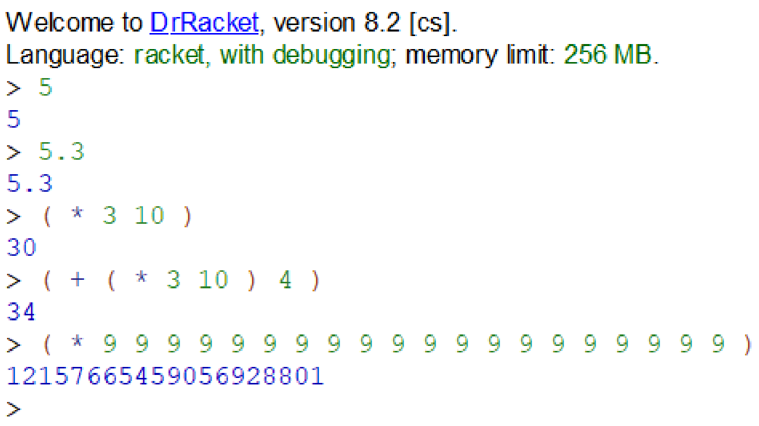
Racket Assignment #2: Interactions, Definitions, Applications

Learning Abstract

This programming assignment is about various Racket mimicking interactions, a number of function definitions, and engaging in computational problem solving that include the reuse of code, imaginative constructions, and the reconfiguration of existing code.

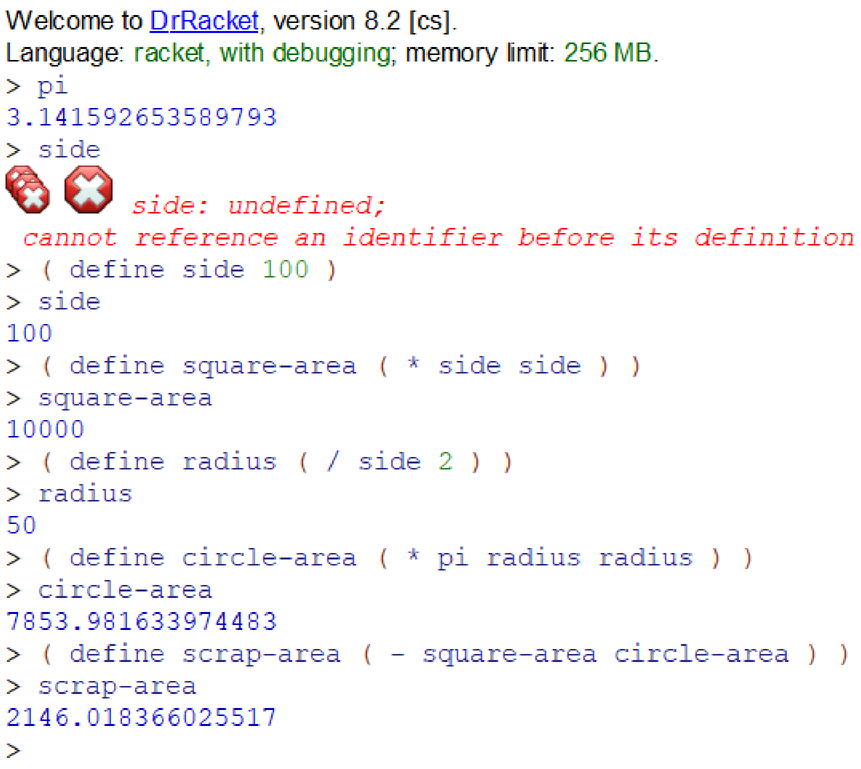
Task 1: Interactions - Scrap of Tin

Arithmetic Expressions:



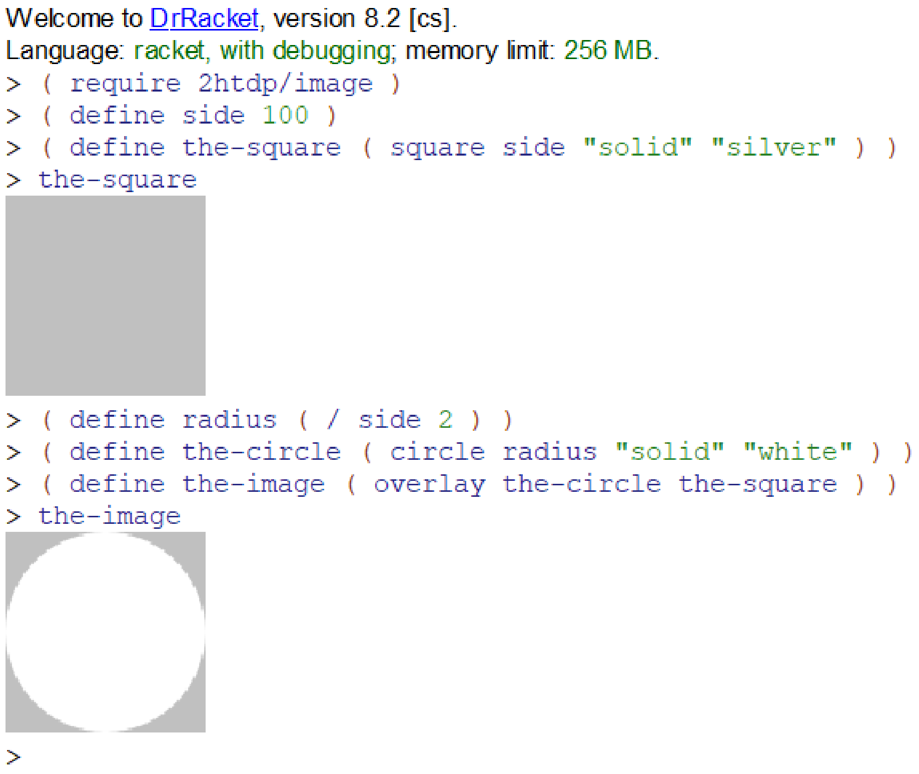
```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 MB.
> 5
5
> 5.3
5.3
> ( * 3 10 )
30
> ( + ( * 3 10 ) 4 )
34
> ( * 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 )
12157665459056928801
>
```

Solving a Simple Problem (Area of Scrap):



```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 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
>
```

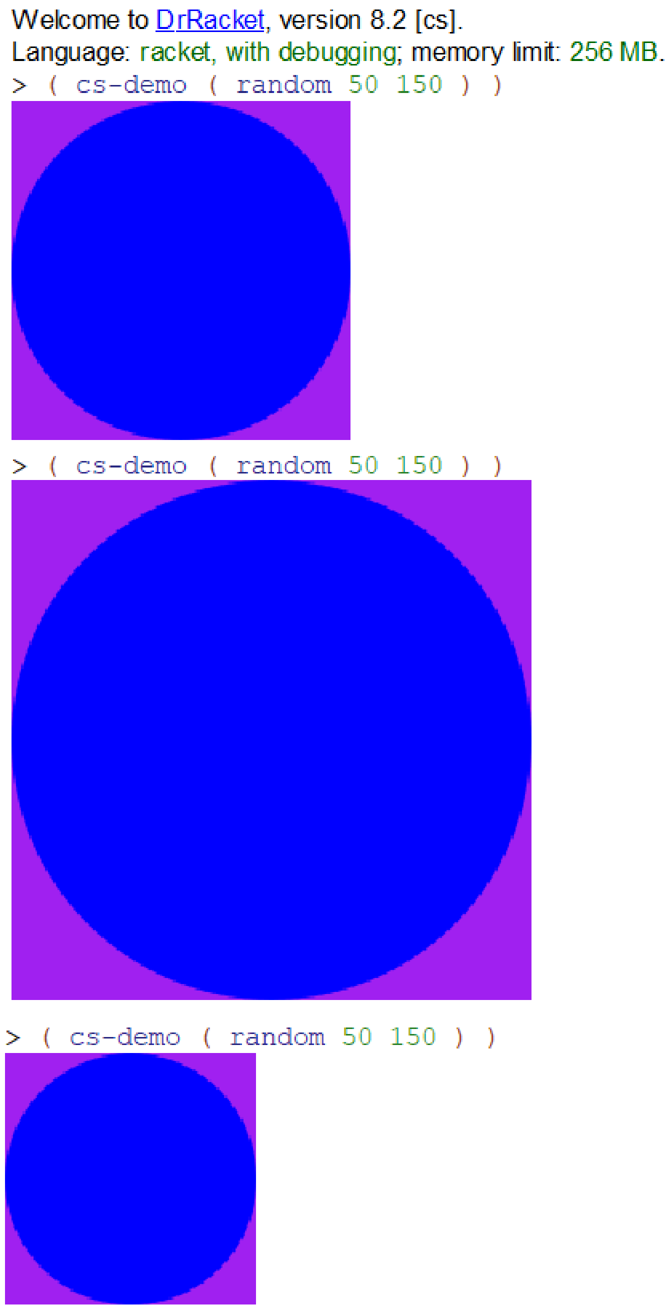
Rendering an Image of the Problem Situation:



```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 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
>
```

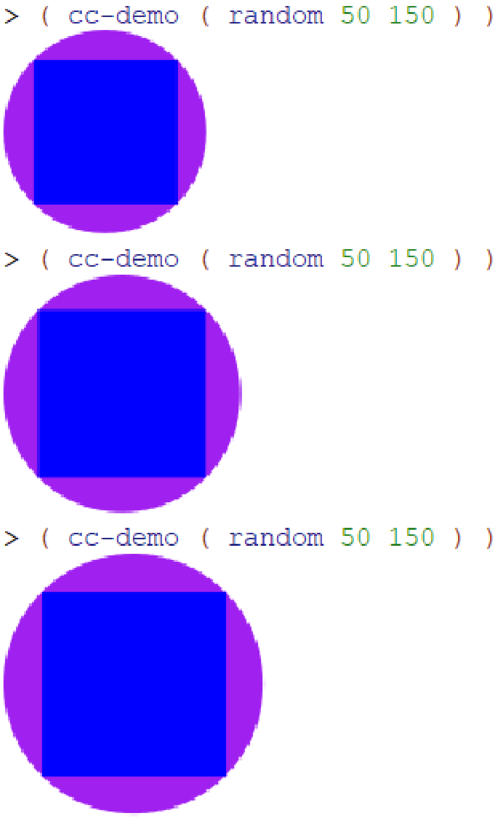
Task 2: Definitions - Inscribing/Circumscribing Circles/Squares

cs-demo:



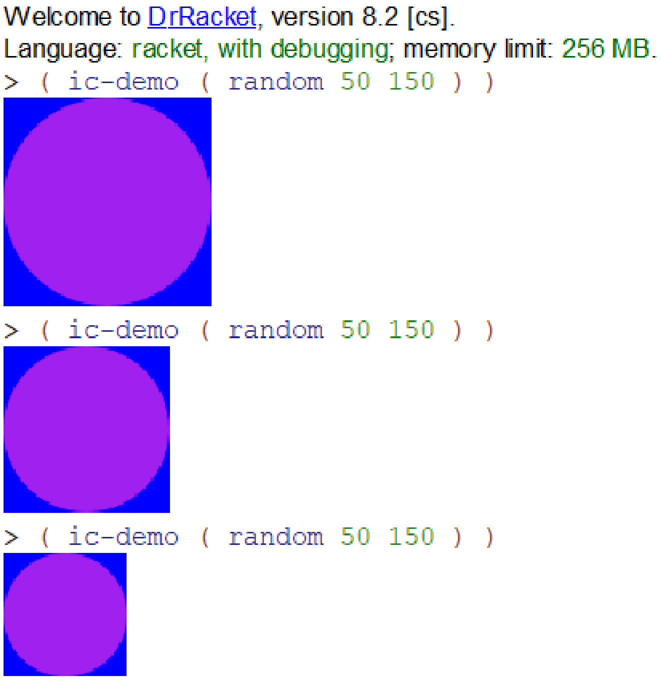
```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 MB.
> ( cs-demo ( random 50 150 ) )
> ( cs-demo ( random 50 150 ) )
> ( cs-demo ( random 50 150 ) )
```

cc-demo



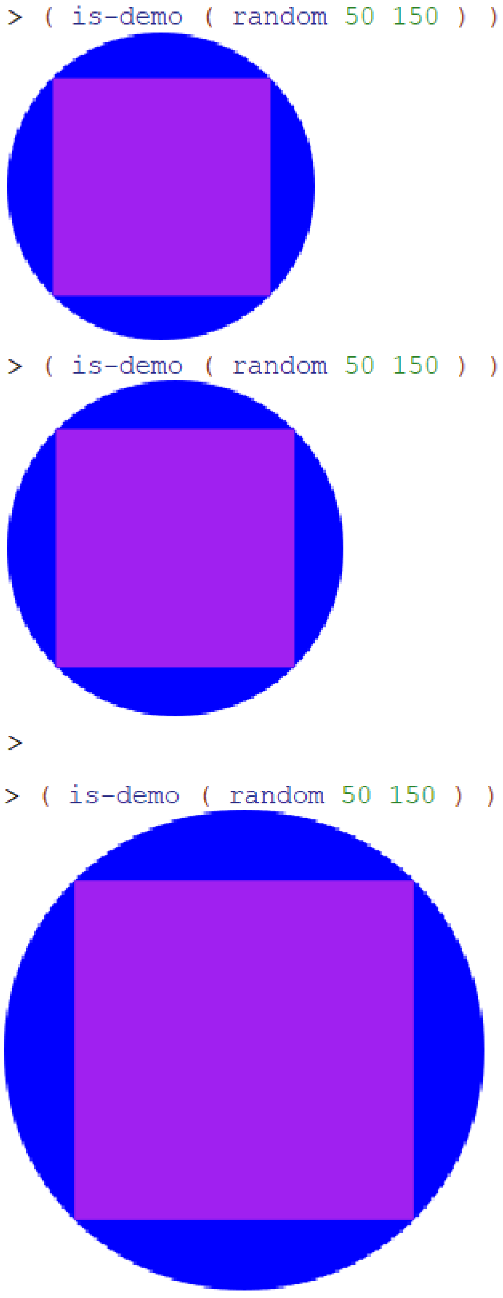
```
> ( cc-demo ( random 50 150 ) )
> ( cc-demo ( random 50 150 ) )
> ( cc-demo ( random 50 150 ) )
```

ic-demo:



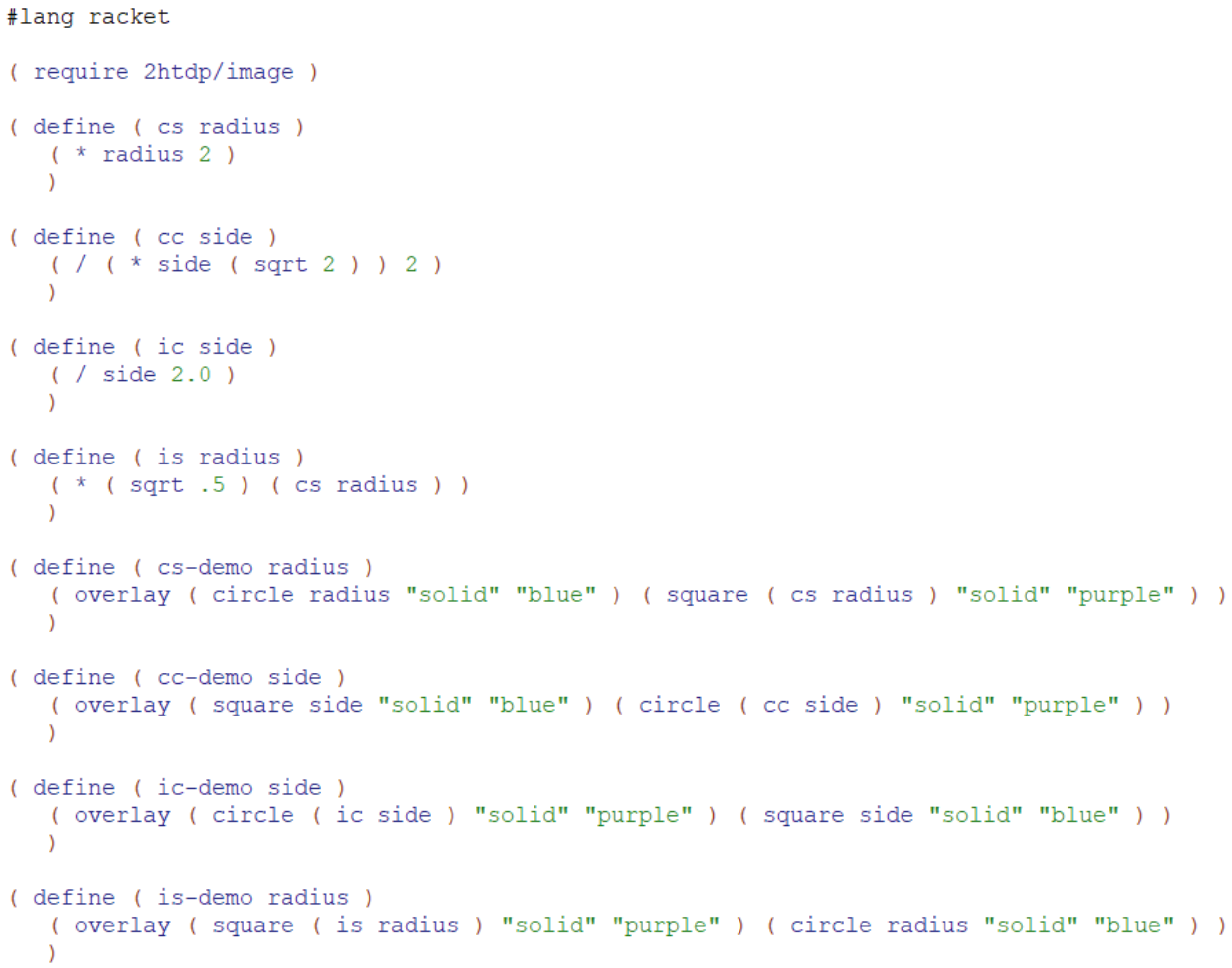
```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 MB.
> ( ic-demo ( random 50 150 ) )
> ( ic-demo ( random 50 150 ) )
> ( ic-demo ( random 50 150 ) )
```

is-demo



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

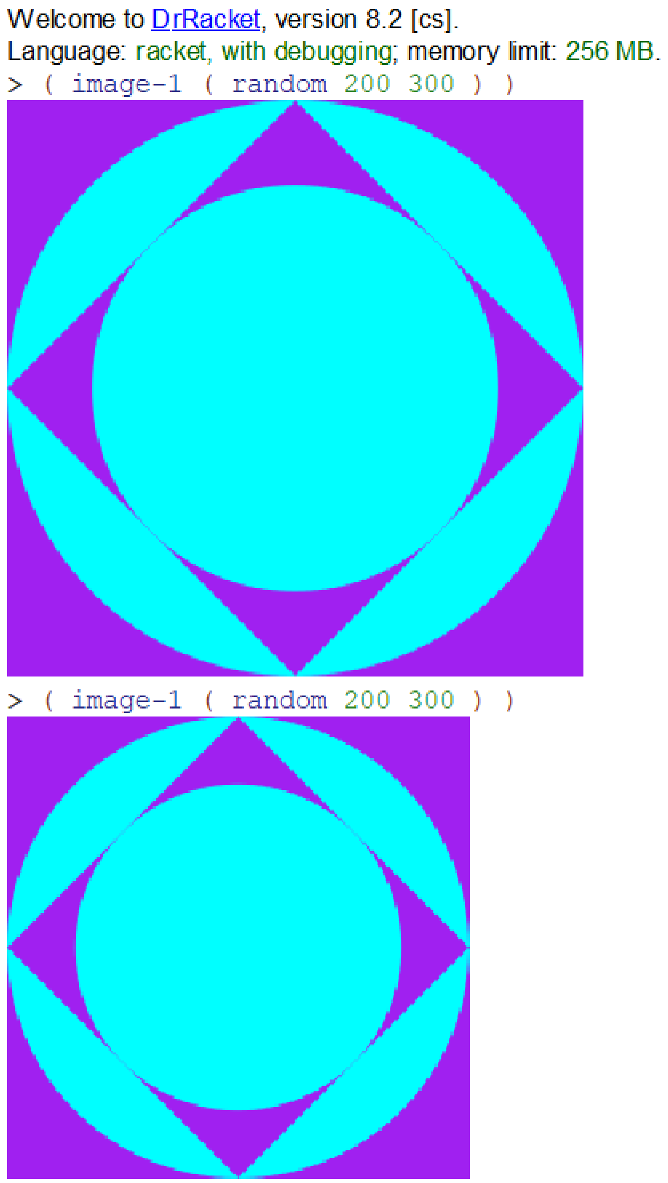
The Code:



```
#lang racket

( require 2htdp/image )

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

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

( define ( ic side )
   ( / side 2.0 )
   )

( define ( is radius )
   ( * ( sqrt .5 ) ( cs radius ) )
   )

( 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: Inscribing/Circumscribing Images

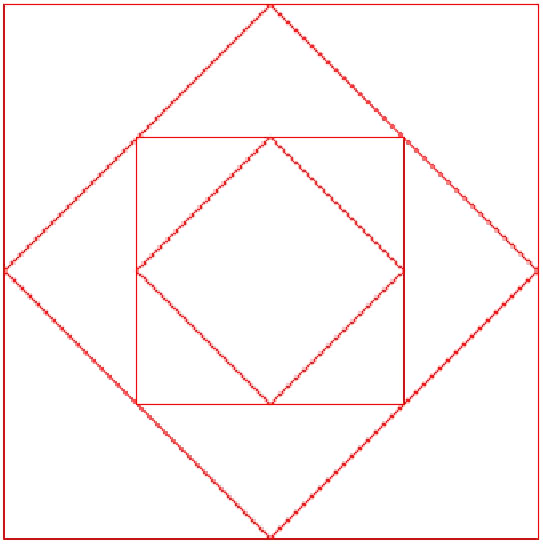
Image 1 Demo:



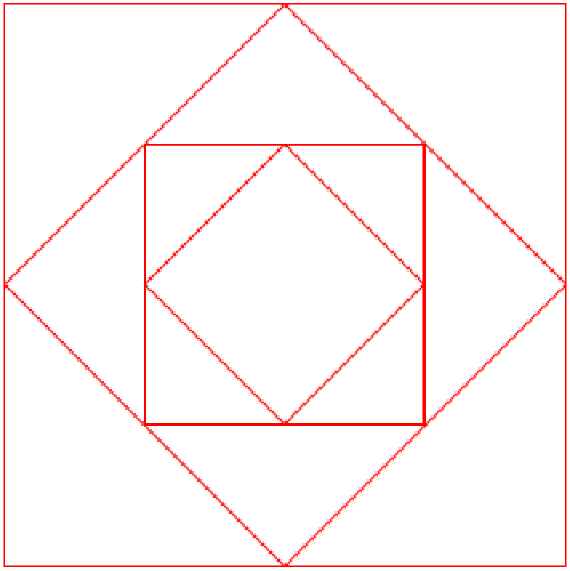
```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 MB.
> ( image-1 ( random 200 300 ) )
> ( image-1 ( random 200 300 ) )
```

Image 2 Demo:

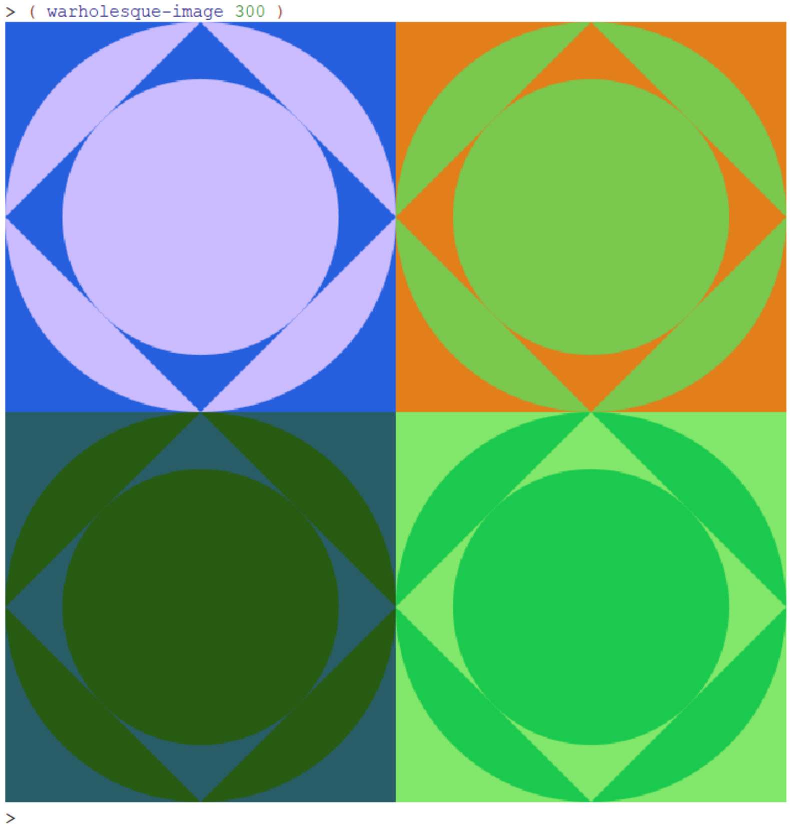
```
> ( image-2 ( random 200 300 ) )
```



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

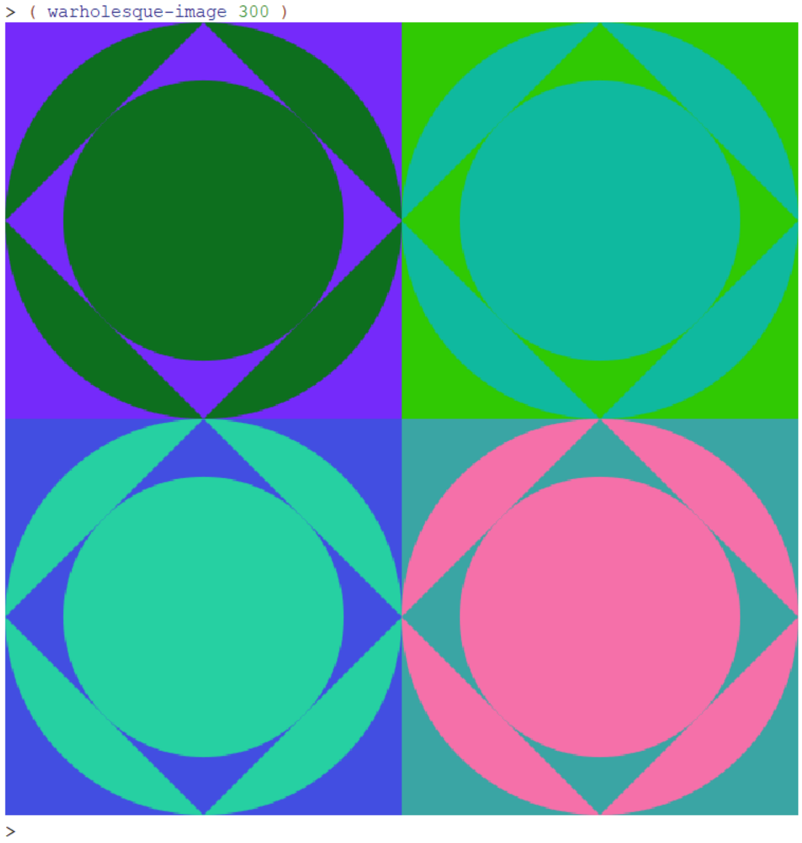


Warholesque image:



```
> ( warholesque-image 300 )
```

```
>
```



```
> ( warholesque-image 300 )
```

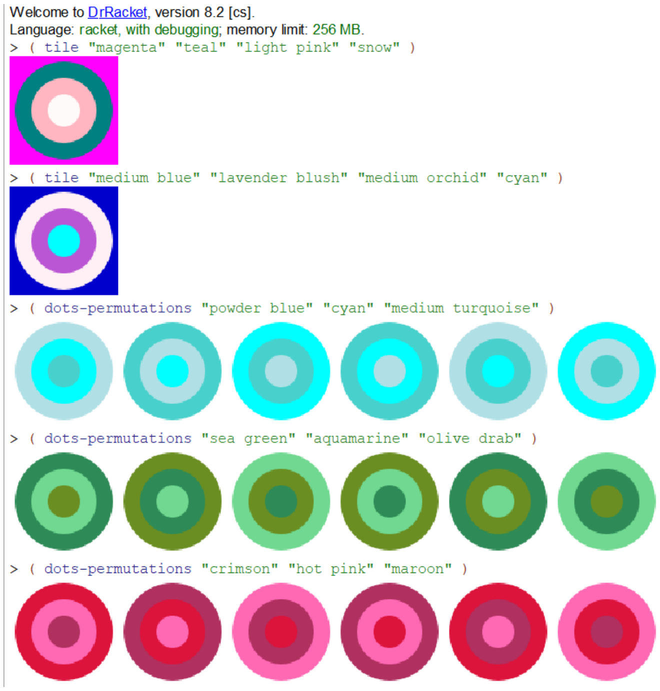
```
>
```

The Code:

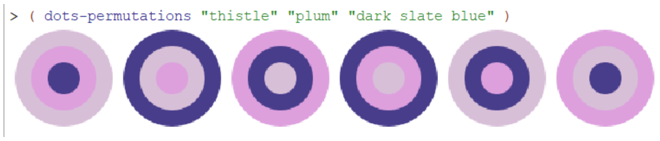
```
( define ( image-1 side )
   ( overlay ( circle ( ic ( is ( ic side ) ) ) "solid" "cyan" )
             ( overlay ( rotate 45 ( square ( is ( ic side ) ) "solid" "purple" ) )
                       ( overlay ( circle ( ic side ) "solid" "cyan" ) ( square side "solid" "purple" ) )
               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

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

( 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 ) )
                       )
             )
   )
```

Task 4: Permutations of Randomly Colored Stacked Dots

Demo:



```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 256 MB.
> ( tile "magenta" "teal" "light pink" "snow" )
> ( tile "medium blue" "lavender blush" "medium orchid" "cyan" )
> ( dots-permutations "powder blue" "cyan" "medium turquoise" )
> ( dots-permutations "sea green" "aquamarine" "olive drab" )
> ( dots-permutations "crimson" "hot pink" "maroon" )
```



```
> ( dots-permutations "thistle" "plum" "dark slate blue" )
```

Code:

```
#lang racket

( require 2htdp/image )

( define ( tile square1 color1 color2 color3 )
   ( define square-tile ( square 100 "solid" square1 ) )
   ( define big-circle (circle 45 "solid" color1 ) )
   ( define middle-circle (circle 30 "solid" color2 ) )
   ( define little-circle (circle 15 "solid" color3 ) )
   ( overlay little-circle middle-circle big-circle square-tile )
   )

( define ( dots-permutations color1 color2 color3 )
   ( beside ( tile "white" color1 color2 color3 )
            ( tile "white" color3 color1 color2 )
            ( tile "white" color2 color3 color1 )
            ( tile "white" color3 color2 color1 )
            ( tile "white" color1 color3 color2 )
            ( tile "white" color2 color1 color3 )
            )
   )
```