Racket Assignment #2 Interactions, Definitions, Applications

Learning Abstract:

This assignment is to explore and showcase problem-solving skills by creating functions using the Racket language.

Task 1: Interactions – Scrap of Tin Arithmetic Expressions

Arithmetic Expressions

Solve a Simple Problem (Area of Scrap)

```
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
>
```

Rendering an Image of the Problem Situation

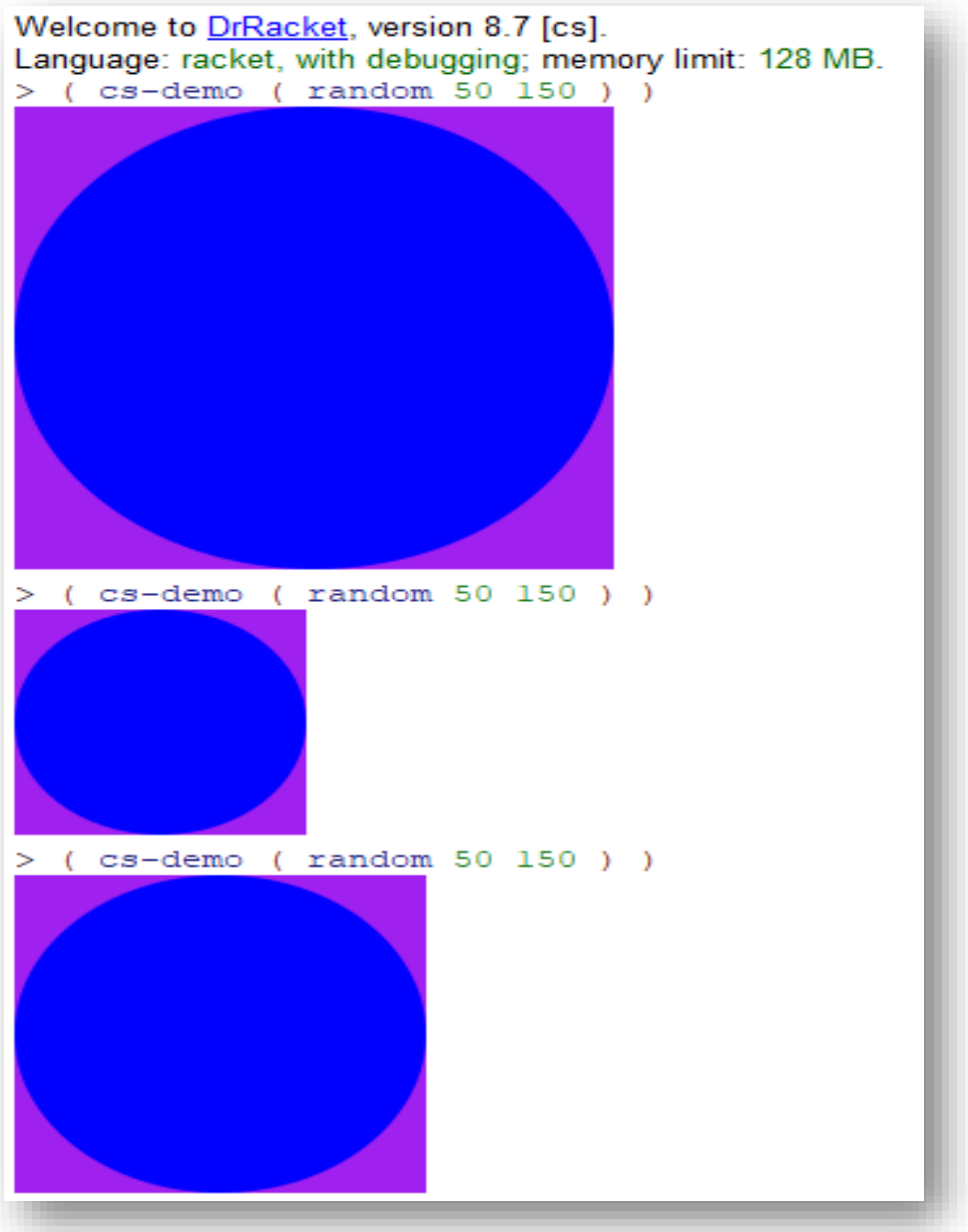
```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( require 2htdp/image )
> ( define side 100 )
> ( define the-square ( square side "solid" "silver" ) )
> the-square
```

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

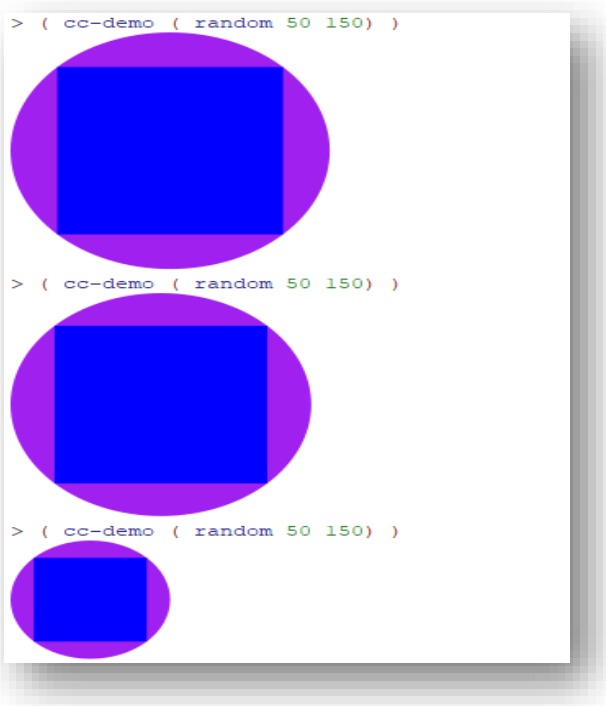
```
>
```

Task 2: Definitions – Inscribing/Circumscribing Circles/Squares

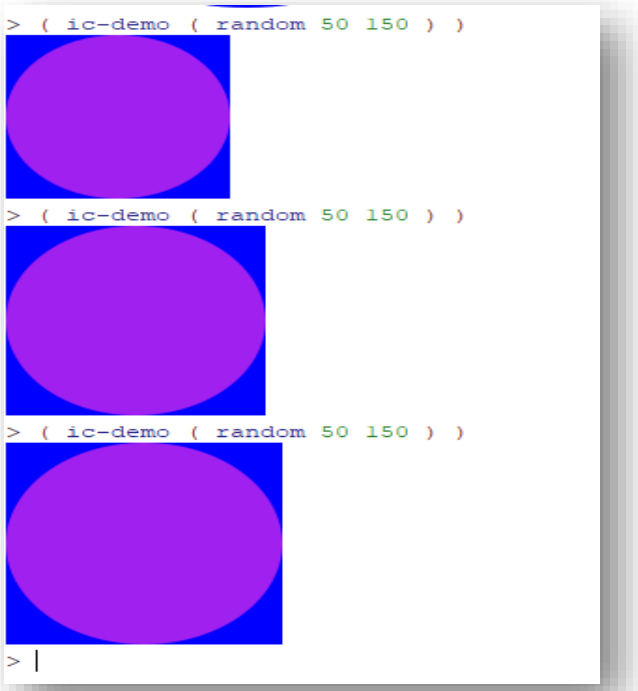
CS-Demo



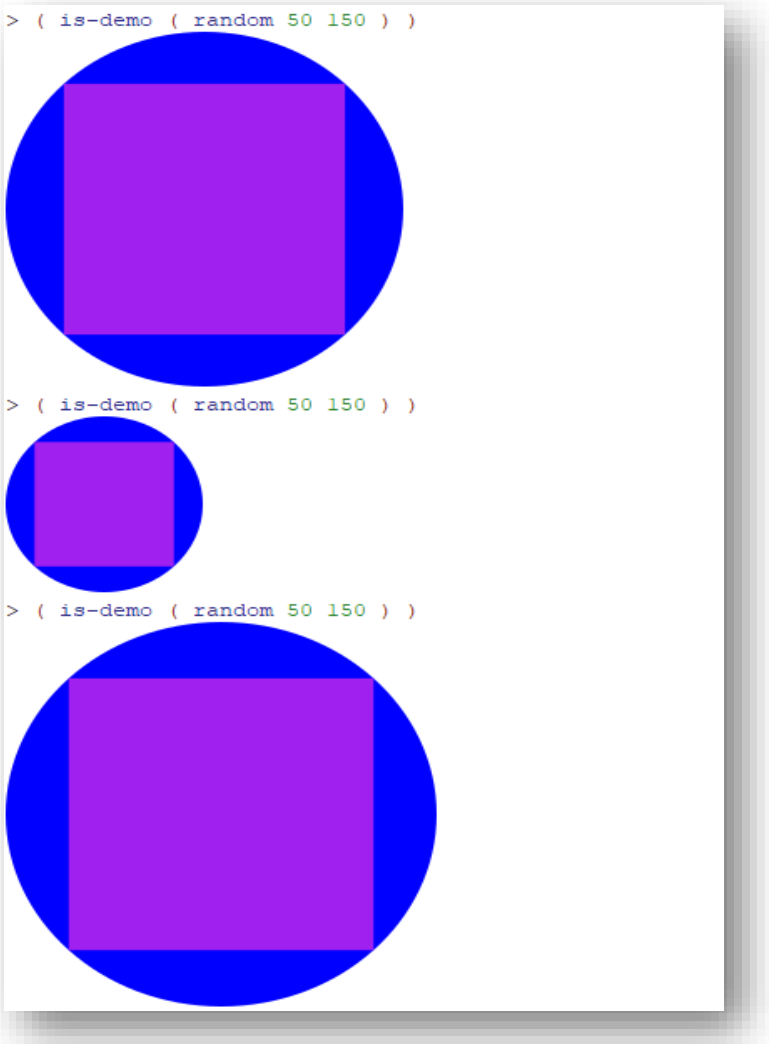
CC-Demo



IC-Demo



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 )
  ( / ( * side-length ( sqrt 2 ) ) 2 )
)
( define ( ic side-length )
  ( / side-length 2 )
)
( define ( is radius )
  ( * radius ( sqrt 2 ) )
)
( define ( cs-demo radius )
  ( define the-square ( square (cs radius) "solid" "purple" ) )
  ( define the-circle ( circle radius "solid" "blue" ) )
  ( overlay the-circle the-square )
)
( define ( cc-demo side-length )
  ( define the-circle ( circle (cc side-length) "solid" "purple" ) )
  ( define the-square ( square side-length "solid" "blue" ) )
  ( overlay the-square the-circle )
)
( define ( ic-demo side-length )
  ( define the-circle ( circle (ic side-length) "solid" "purple" ) )
  ( define the-square ( square side-length "solid" "blue" ) )
  ( overlay the-circle the-square )
)
( define ( is-demo radius )
  ( define the-square ( square (is radius) "solid" "purple" ) )
  ( define the-circle ( circle radius "solid" "blue" ) )
  ( overlay the-square the-circle )
)
```

Task 3: Inscribing/Circumscribing Images

Image 1 Demo

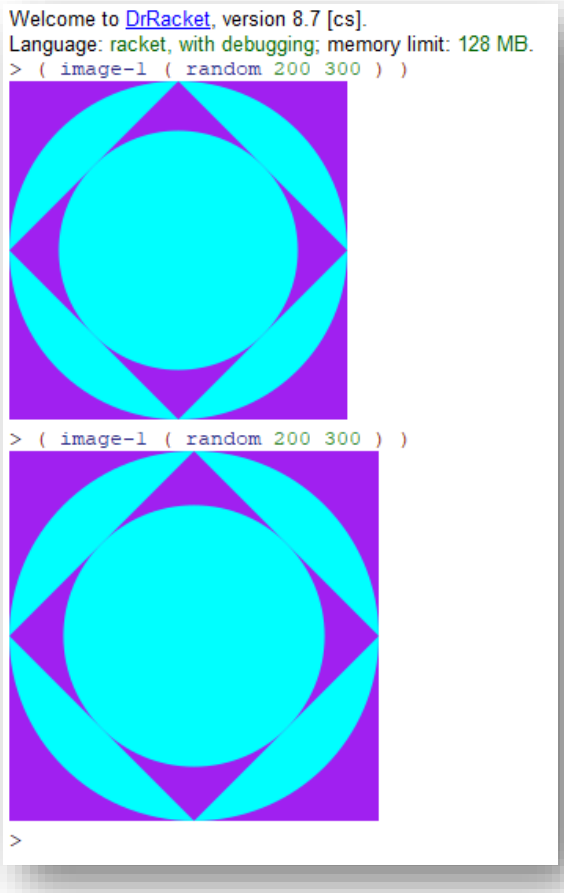
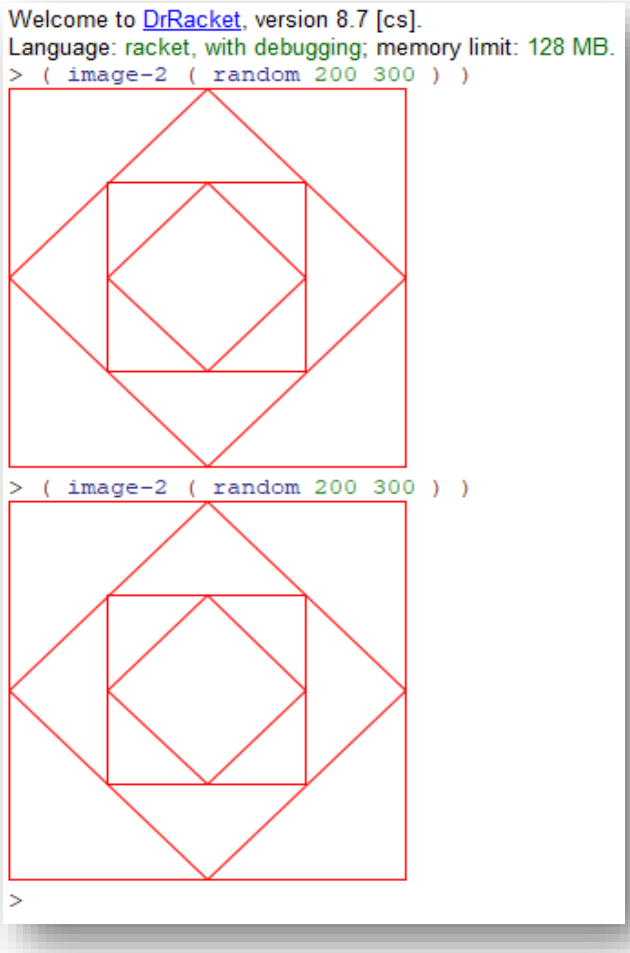
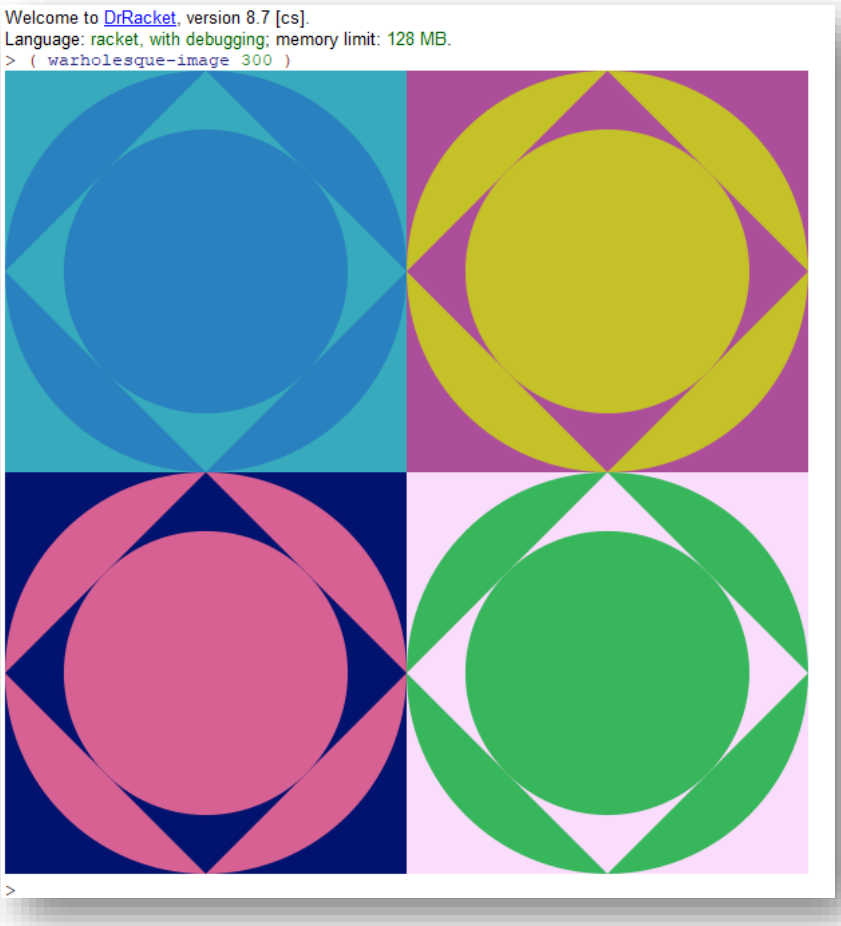


Image 2 Demo



Warholesque Image



The Code

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


( define ( image-2 side-length )
 ( define the-square2 (is ( ic side-length ) ) )
 ( define the-square3 (is ( ic the-square2 ) ) )
 ( define the-square4 (is ( ic the-square3 ) ) )
 ( overlay ( rotate 45 ( square the-square4 "outline" "red" ) )
 ( overlay ( square the-square3 "outline" "red" )
 ( overlay ( rotate 45 ( square the-square2 "outline" "red" ) )
 ( overlay ( square side-length "outline" "red" )
 ( square side-length "outline" "red" ) ) ) ) )
 )

( define ( warholesque-single-image canvas-length)
 ( define ( rbg ) ( random 0 256 ) )
 ( define ( rc ) ( color ( rbg ) ( rbg ) ( rbg ) ) )
 ( define c1 ( rc ) )
 ( define c2 ( rc ) )
 ( overlay ( circle ( ic ( is ( ic canvas-length ) ) ) "solid" c2 )
 ( overlay ( rotate 45 ( square ( is ( ic canvas-length ) ) "solid" c1 ) )
 ( overlay ( circle ( ic canvas-length ) "solid" c2 ) ( square canvas-length "solid" c1 ) ) ) )
)

( define ( warholesque-image canvas-length )
( above
 ( beside
   ( warholesque-single-image canvas-length )
   ( warholesque-single-image canvas-length )
  )
 ( beside
   ( warholesque-single-image canvas-length )
   ( warholesque-single-image canvas-length )
  )
 )
)
```

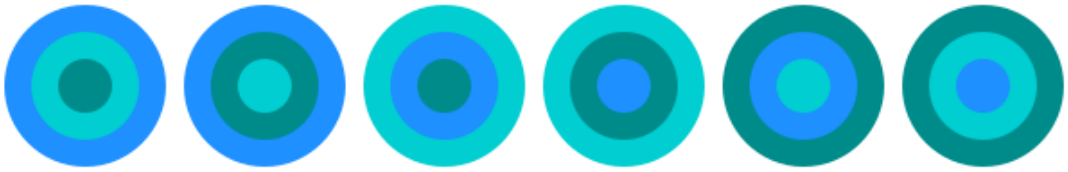
Task 4: Permutations of Randomly Colored Stacked Dots

Demo

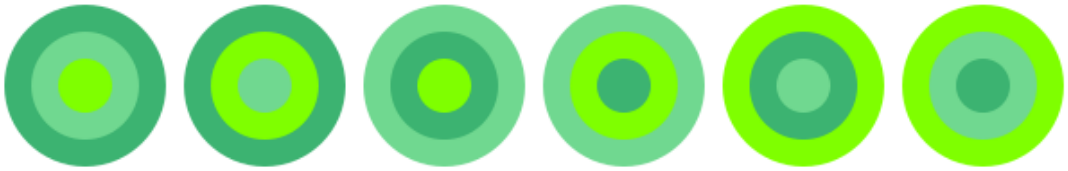
```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( tile "Indian Red" "Peru" "Orchid" "Steel Blue" )
```

```
> ( tile "Black" "Dark Magenta" "Blue Violet" "Pale Turquoise" )
```

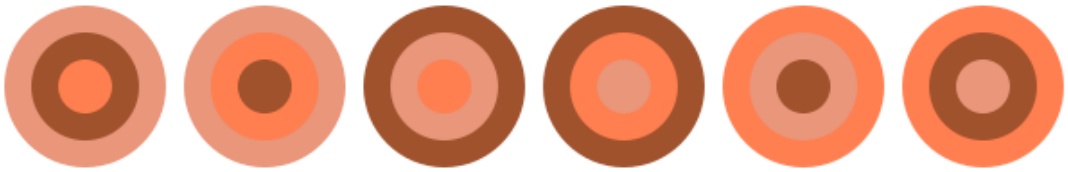
```
> ( dots-permutations "Dodger Blue" "Dark Turquoise" "Dark cyan" )
```



```
> ( dots-permutations "Medium Sea Green" "Aquamarine" "Chartreuse" )
```



```
> ( dots-permutations "Dark Salmon" "Sienna" "Coral" )
```



The Code

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

( define ( tile square-color c1 c2 c3 )
  ( define square-tile( square 100 "solid" square-color ) )
  ( define circle1 ( circle ( / 90 2 ) "solid" c1 ) )
  ( define circle2 ( circle ( / 60 2 ) "solid" c2 ) )
  ( define circle3 ( circle ( / 30 2 ) "solid" c3 ) )
  ( define small ( overlay circle3 circle2 ) )
  ( define big ( overlay circle1 square-tile ) )
  ( overlay small big )
)

( define ( dots-permutations c1 c2 c3 )
  ( define circle1 ( tile "white" c1 c2 c3 ) )
  ( define circle2 ( tile "white" c1 c3 c2 ) )
  ( define circle3 ( tile "white" c2 c1 c3 ) )
  ( define circle4 ( tile "white" c2 c3 c1 ) )
  ( define circle5 ( tile "white" c3 c1 c2 ) )
  ( define circle6 ( tile "white" c3 c2 c1 ) )
  ( beside
   ( beside
    ( beside
     ( beside
      ( beside circle1 circle2 )
      circle3 )
     circle4 )
    circle5 )
   circle6 )
)
```