
Racket Assignment #2: Interactions, Definitions, Applications

ABSTRACT:

For the second Racket assignment, the main objective was to explore basic Racket programming concepts through four distinct tasks. The first task works with arithmetic expressions and generating shapes. The second and third tasks involve defining and generating inscribing and circumscribing images. The fourth task focuses on generating permutations of randomly generated colored dots. Throughout these tasks, the assignment emphasizes numeric processing and the creation of colored, inscribing or circumscribing images.

Task 1: Interactions - Scrap of Tin

Arithmetic Expressions

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 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 )
121576654590569288O1
>
```

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
> ( 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" "gold" ) )
> the-square
```



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

```
>
```

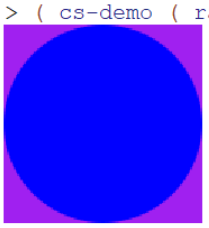
Task 2: Definitions - Inscribing/Circumscribing Circles/Squares

---

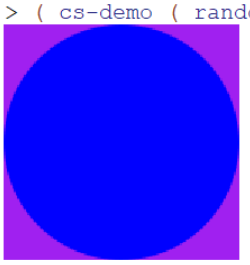
cs-demo

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

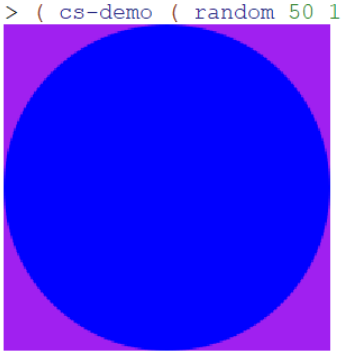
```
> ( cs-demo ( random 50 150 ) )
```



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



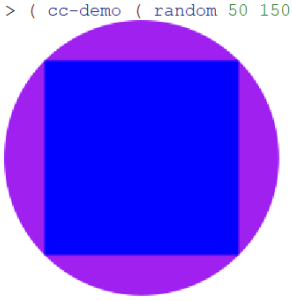
```
> ( cs-demo ( random 50 150 ) )
```



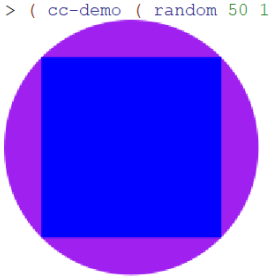
```
>
```

cc-demo

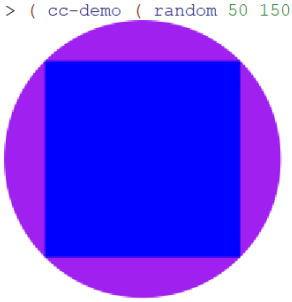
```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( cc-demo ( random 50 150 ) )
```



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



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

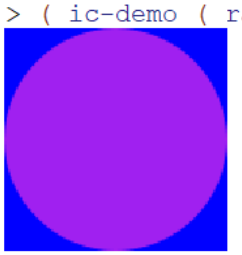


```
>
```

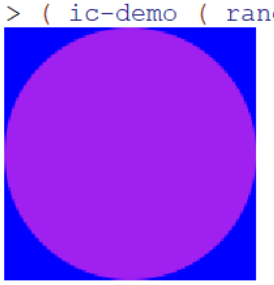
ic-demo

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

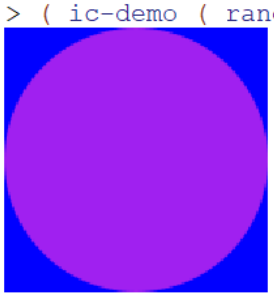
```
> ( ic-demo ( random 50 150 ) )
```



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



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

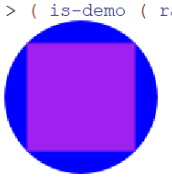


```
>
```

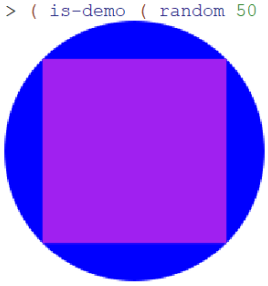
is-demo

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

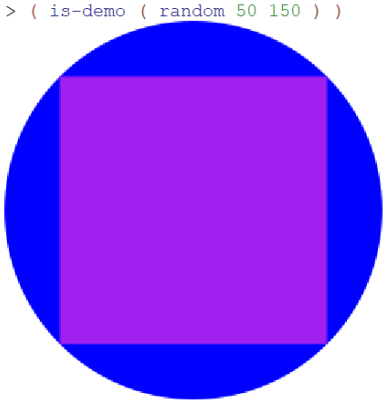
```
> ( is-demo ( random 50 150 ) )
```



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



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



```
>
```

The Code

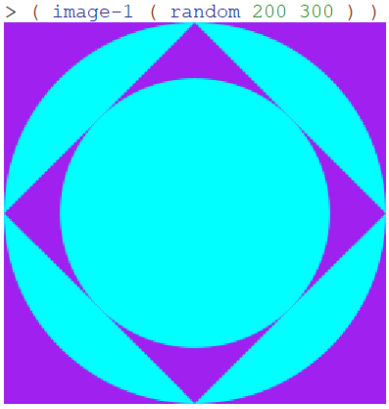
```
#lang racket
( require 2htdp/image )


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

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

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

( define ( is radius )
   ( * ( sqrt 0.5 ) ( cs radius ) )
)

( define ( cs-demo radius )
   ( define ci( circle radius "solid" "blue" ) )
   ( define sq ( square ( cs radius ) "solid" "purple" ) )
   ( overlay ci sq )
)

( define ( cc-demo side-length )
   ( define sq ( square side-length "solid" "blue" ) )
   ( define ci( circle ( cc side-length ) "solid" "purple" ) )
   ( overlay sq ci )
)

( define ( ic-demo side-length )
   ( define ci( circle ( ic side-length ) "solid" "purple" ) )
   ( define sq ( square side-length "solid" "blue" ) )
   ( overlay ci sq )
)

( define ( is-demo radius )
   ( define sq ( square ( is radius ) "solid" "purple" ) )
   ( define ci( circle radius "solid" "blue" ) )
   ( overlay sq ci )
)
```

Task 3: Inscribing/Circumscribing Images

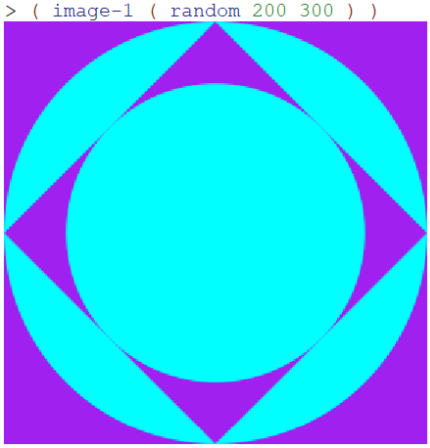
Image 1 Demo

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

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



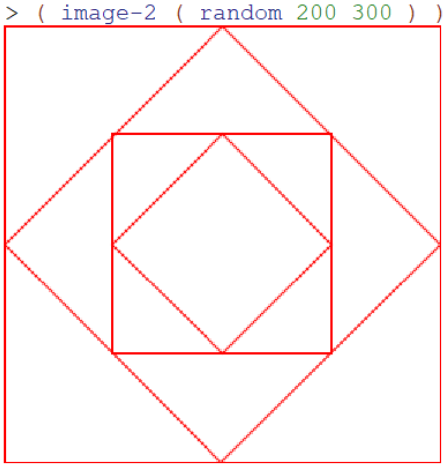
```
> ( image-1 ( random 200 300 ) )
```



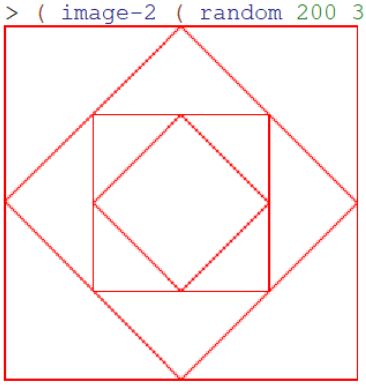
```
>
```

Image 2 Demo

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( image-2 ( random 200 300 ) )
```



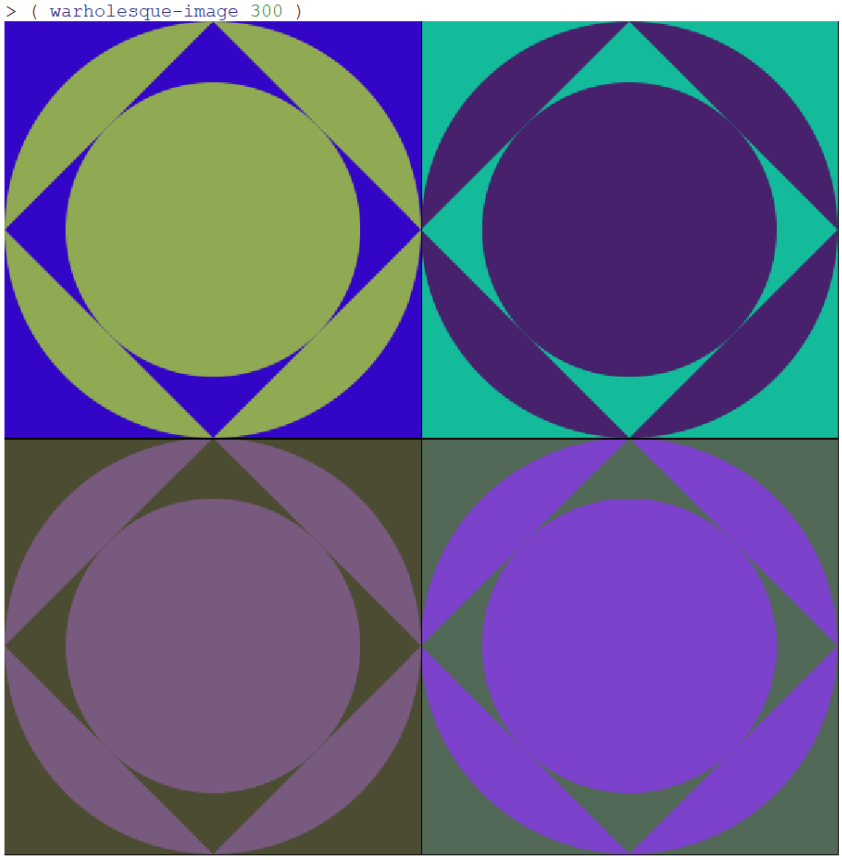
```
> ( image-2 ( random 200 300 ) )
```



```
>
```

Warholesque Image

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



```
>
```

The Code

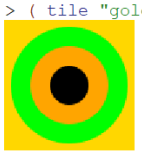
```
49
50 ( define ( image-1 side-length )
51    ( overlay ( circle ( ic ( is ( ic side-length ) ) ) "solid" "cyan" )
52    ( overlay ( rotate 45 ( square ( is ( ic side-length ) ) "solid" "purple" ) )
53    ( overlay ( circle ( ic side-length ) "solid" "cyan" ) ( square side-length "solid" "purple" ) ) ) )
54 )
55
56 ( define ( image-2 side-length )
57    ( define box-two (is ( ic side-length ) ) )
58    ( define box-three (is ( ic box-two ) ) )
59    ( define box-four (is ( ic box-three ) ) )
60    ( overlay ( rotate 45 ( square box-four "outline" "red" ) )
61    ( overlay ( square box-three "outline" "red" )
62    ( overlay ( rotate 45 ( square box-two "outline" "red" ) )
63    ( overlay ( square side-length "outline" "red" )
64    ( square side-length "outline" "red" ) ) ) ) )
65  )
66
67
68 ( define ( individual-image box-length )
69    ( define ( rgb ) ( random 0 256 ) )
70    ( define ( random-color ) ( color ( rgb ) ( rgb ) ( rgb ) ) )
71    ( define color1 ( random-color ) )
72    ( define color2 ( random-color ) )
73    ( overlay ( circle ( ic ( is ( ic box-length ) ) ) "solid" color2 )
74    ( overlay ( rotate 45 ( square ( is ( ic box-length ) ) "solid" color1  ) )
75    ( overlay ( circle ( ic box-length ) "solid" color2 ) ( square box-length "solid" color1 ) ( square box-length "outline" "black" ) ) ) )
76 )
77
78 ( define ( warholesque-image image-length )
79    ( above
80       ( beside
81         ( individual-image image-length )
82         ( individual-image image-length )
83       )
84
85       ( beside
86         ( individual-image image-length )
87         ( individual-image image-length )
88       )
89    )
90 )
```

Task 4: Permutations of Randomly Colored Stacked Dots

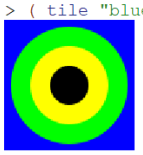
Demo

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

```
> ( tile "gold" "green" "orange" "black" )
```



```
> ( tile "blue" "green" "yellow" "black" )
```

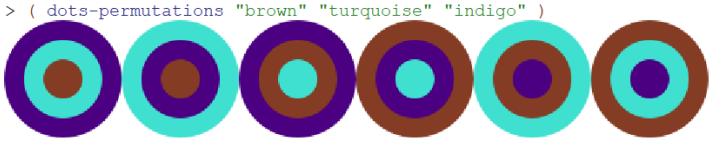


```
> ( dots-permutations "red" "olive" "black" )
```

```
> ( dots-permutations "blue" "salmon" "gold" )
```

```
> ( dots-permutations "violet" "black" "orange" )
```

```
> ( dots-permutations "brown" "turquoise" "indigo" )
```



```
>
```

Code

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

( define ( tile color1 color2 color3 color4)
 ( overlay ( circle 15 "solid" color4 )
 ( overlay ( circle 30 "solid" color3 )
 ( overlay ( circle 45 "solid" color2 )
 (square 100 "solid" color1) ) ) )
 )

( define ( single-dot color1 color2 color3 )
 ( overlay ( circle 15 "solid" color1 )
 ( overlay ( circle 30 "solid" color2 )
 ( circle 45 "solid" color3 ) ) )
)

( define ( dots-permutations color1 color2 color3 )
   ( beside
     ( single-dot color1 color2 color3 )
   ( beside
     ( single-dot color1 color3 color2 )
   ( beside
     ( single-dot color2 color1 color3 )
   ( beside
     ( single-dot color2 color3 color1 )
   ( beside
     ( single-dot color3 color1 color2 )
     ( single-dot color3 color2 color1 ) ) ) ) ) )
)

```