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 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
> 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

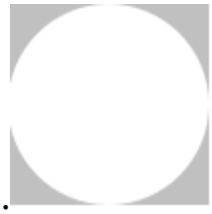
```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( require 2htdp/image )
> ( define side 100 )
> ( define the-square ( square side "solid" "silver" ) )
> the-square
```

```
> ( define radius ( / side 2 ) )
```

> (define the-circle (circle radius "solid" "white"))

> (define the-image (overlay the-circle the-square))

> the-image



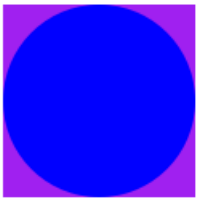
.

>

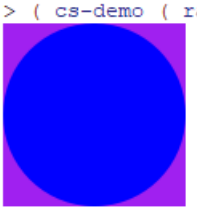
Task 2: Definitions - Inscribing/Circumscribing Circles/Squares

cs-demo

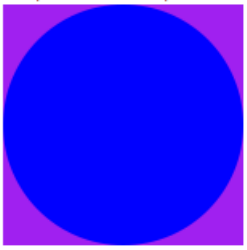
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (cs-demo (random 50 150))



> (cs-demo (random 50 150))



> (cs-demo (random 50 150))

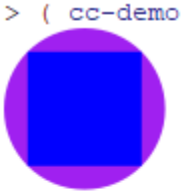


>

cc-demo

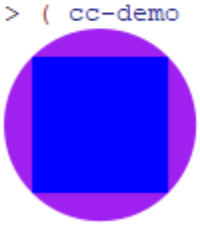
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.

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



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

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

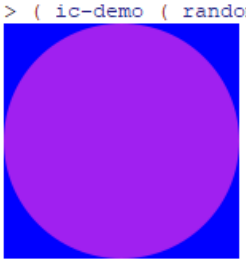


```
>
```

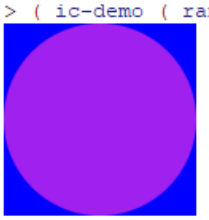
ic-demo

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

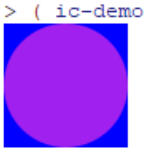
```
> ( ic-demo ( random 50 150 ) )
```



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



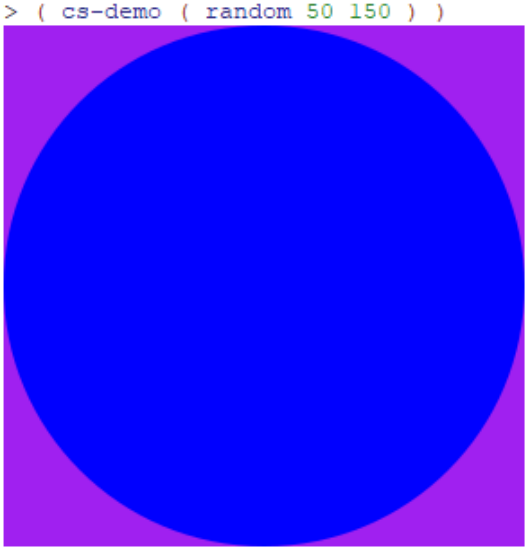
```
> ( ic-demo ( random 50 150 ) )
```



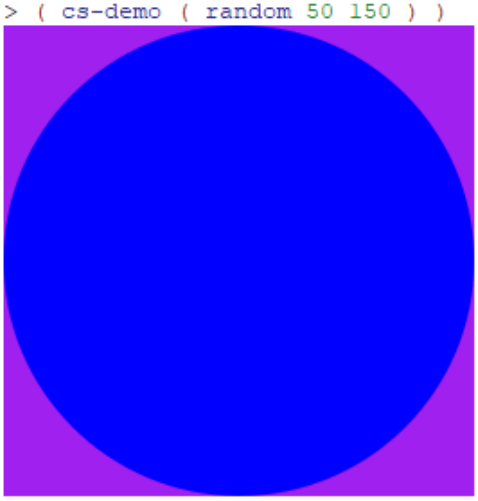
```
>
```

is-demo

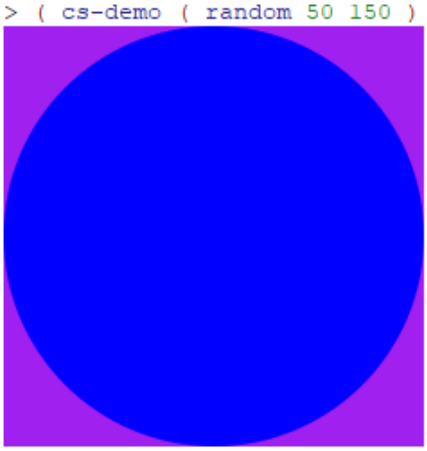
```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( cs-demo ( random 50 150 ) )
```



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



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



```
>
```

The Code

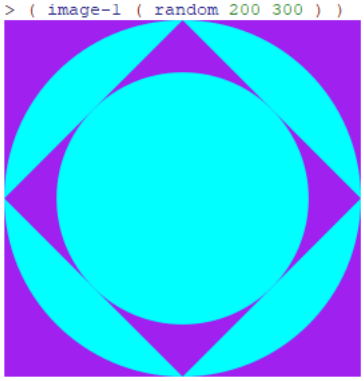
```
#lang racket

( require 2htdp/image )


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


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


( define ( ic side-length)
  (exact->inexact (/ side-length 2))
  )


( 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-length)
  (  overlay ( square side-length "solid" "blue" ) (circle (cc side-length) "solid" "purple") )
```

```
  )

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

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

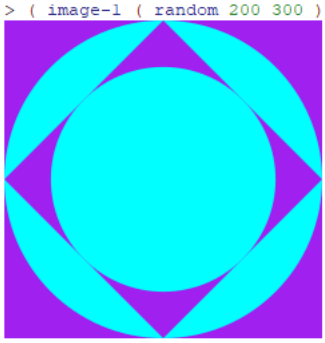
Task 3: Inscribing/Circumscribing Images

Image 1 Demo

```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( image-1 ( random 200 300 ) )
```



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

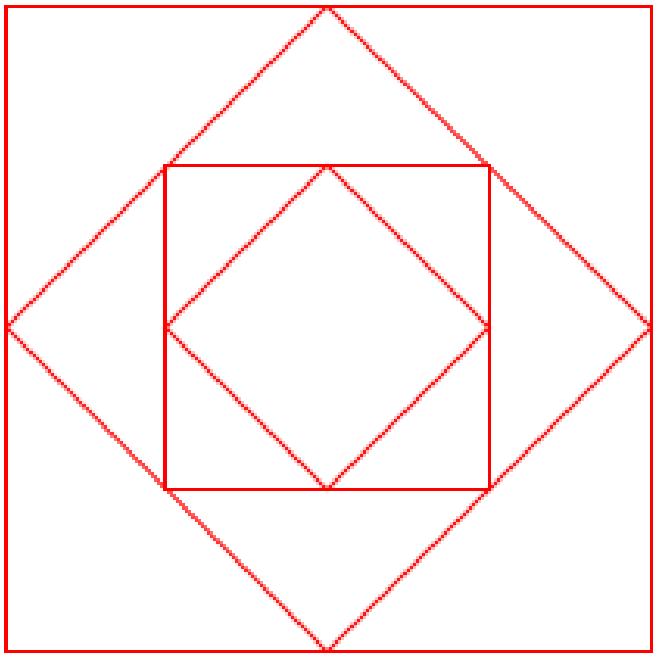


```
>
```

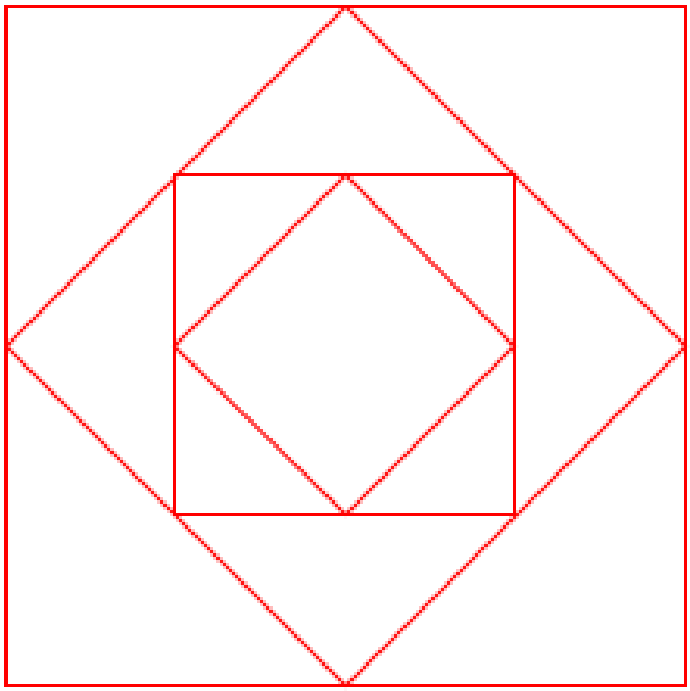
Image 2 Demo

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

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



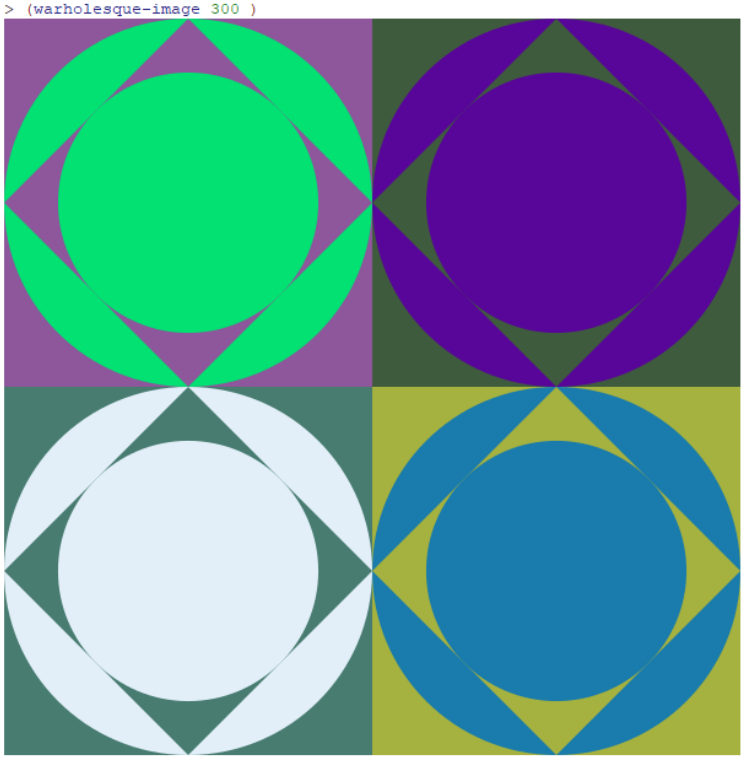
```
> (image-2 ( random 200 300 ) )
```



```
>
```

Warholesque Image

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



```
>
```

The Code

```
( 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 second-box (is ( ic side-length ) ) )
  ( define third-box (is ( ic second-box ) ) )
  ( define fourth-box (is ( ic third-box ) ) )
```

```
  ( overlay ( rotate 45 ( square fourth-box "outline" "red" ) )
  ( overlay ( square third-box "outline" "red" )
  ( overlay ( rotate 45 ( square second-box "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
```

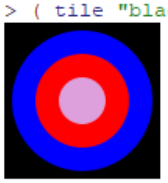
```
    ( warholesque-single-image canvas-length )
    ( warholesque-single-image canvas-length )
  )
 )
)
```

Task 4: Permutations of Randomly Colored Stacked Dots

Demo

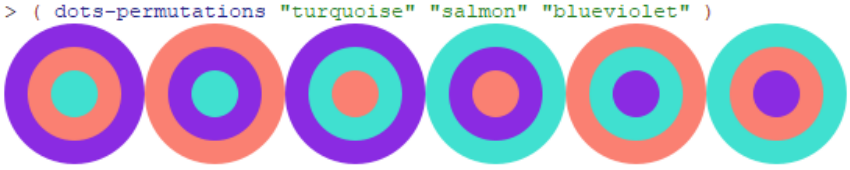
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.

```
> ( tile "black" "blue" "red" "plum" )
```



```
> ( tile "brown" "magenta" "black" "yellow" )
```

```
> ( dots-permutations "turquoise" "salmon" "blueviolet" )
```



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

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

```
> ( dots-permutations "black" "cornflowerblue" "teal" )
```

```
>
```

Code

```
#lang racket

( require 2htdp/image )


( define ( tile c1 c2 c3 c4)
    ( overlay ( circle 15 "solid" c4 )
    ( overlay ( circle 30 "solid" c3 )
    ( overlay ( circle 45 "solid" c2 )
    (square 100 "solid" c1) ) ) )
   )


( define ( dot-permutation c1 c2 c3 )
    ( overlay ( circle 15 "solid" c1 )
    ( overlay ( circle 30 "solid" c2 )
     ( circle 45 "solid" c3 )
    )
   )
  )


( define ( dots-permutations  c1 c2 c3 )
   ( beside
        ( dot-permutation c1 c2 c3 )
    ( beside
        ( dot-permutation c1 c3 c2 )
     ( beside
        ( dot-permutation c2 c1 c3 )
```

```
( beside
  ( dot-permutation c2 c3 c1 )
 ( beside
  ( dot-permutation c3 c1 c2 )
  ( dot-permutation c3 c2 c1 )
    )
  )
 )
)
)
```