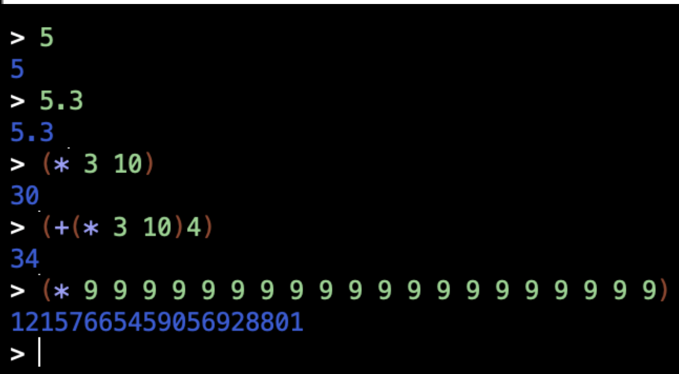
CSC 344 First Racket Programming Assignment Solution

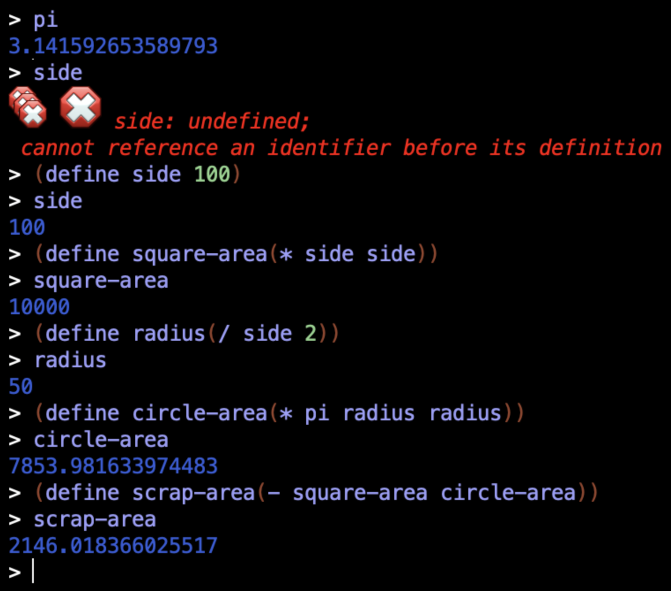
Learning Abstract: This assignment touches on numerical computations as well as visual representations in the DrRacket environment while using the Racket programming language.

First Task: Simple Numerical Processing



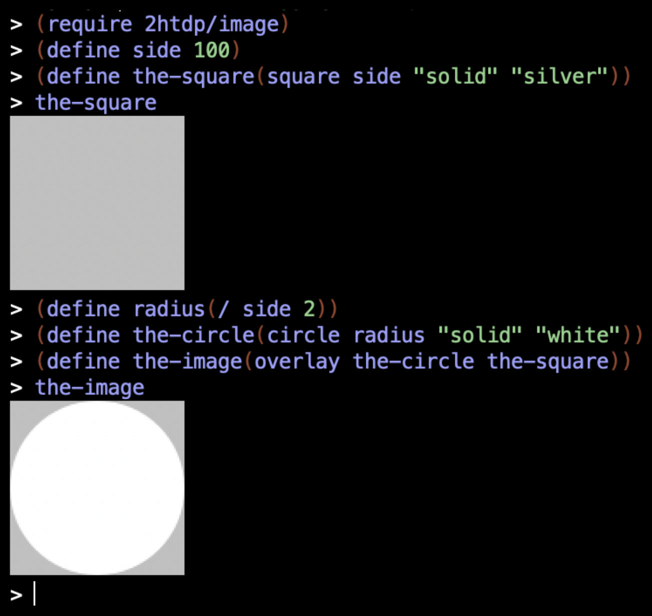
```
> 5
5
> 5.3
5.3
> (* 3 10)
30
> (+(* 3 10)4)
34
> (* 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9)
12157665459056928801
> |
```

Second Task: Solution to the Scrap Problem



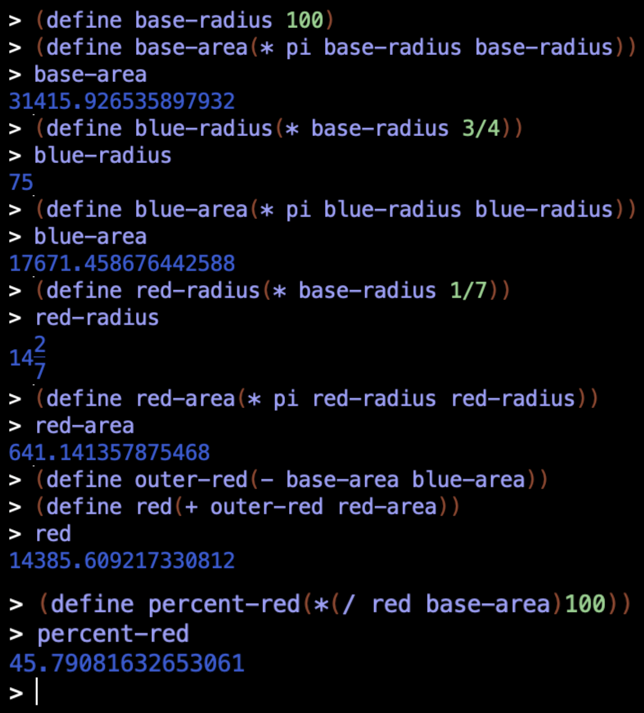
```
> pi
3.141592653589793
> side
      side: undefined;
 cannot reference an identifier before its definition
> (define side 100)
> side
100
> (define square-area(* side side))
> square-area
10000
> (define radius(/ side 2))
> radius
50
> (define circle-area(* pi radius radius))
> circle-area
7853.981633974483
> (define scrap-area(- square-area circle-area))
> scrap-area
2146.018366025517
> |
```

Third Task: Illustration of the Scrap Problem



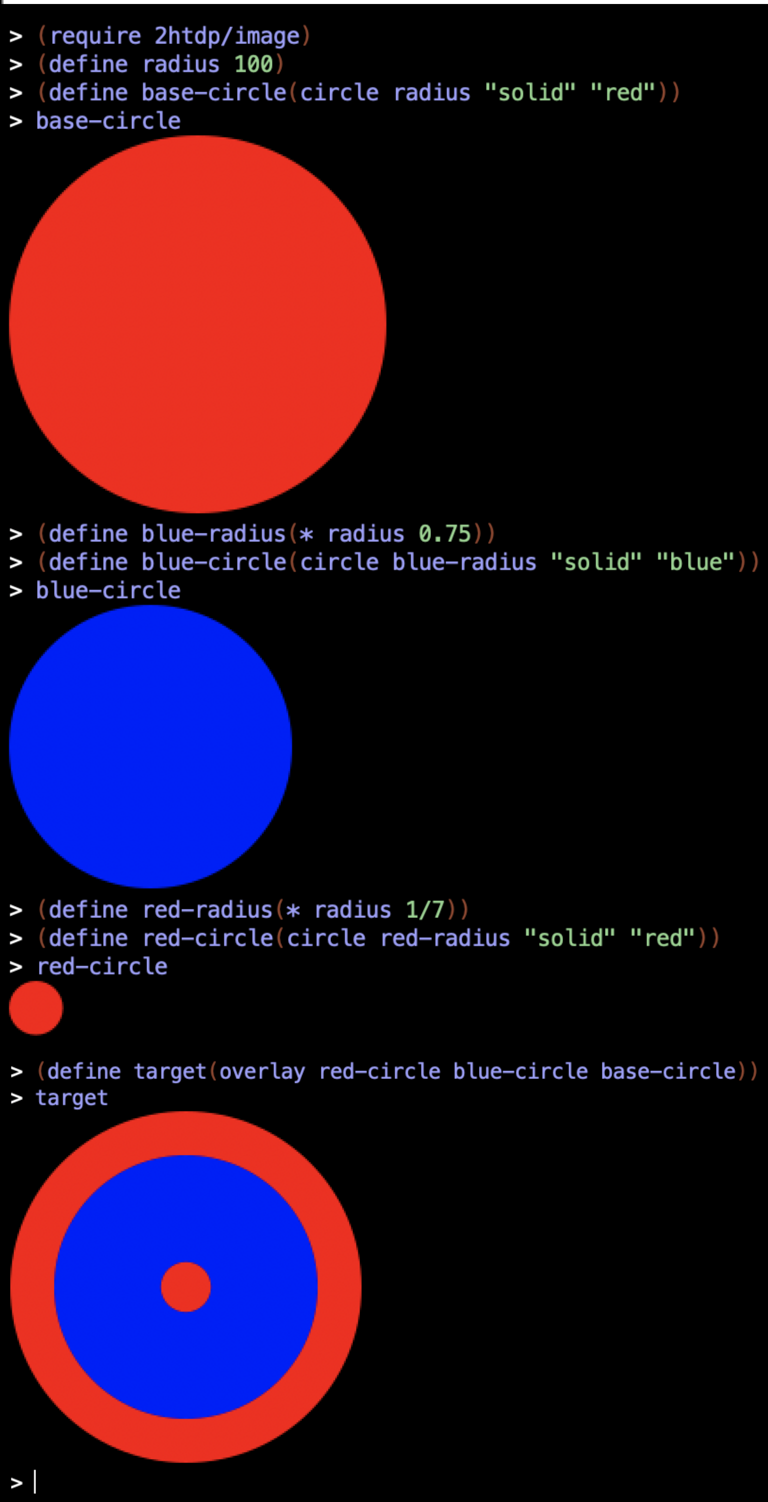
```
> (require 2htdp/image)
> (define side 100)
> (define the-square(square side "solid" "silver"))
> the-square
> (define radius(/ side 2))
> (define the-circle(circle radius "solid" "white"))
> (define the-image(overlay the-circle the-square))
> the-image
> |
```

Fourth Task: Solution to the Target Problem



```
> (define base-radius 100)
> (define base-area(* pi base-radius base-radius))
> base-area
31415.926535897932
> (define blue-radius(* base-radius 3/4))
> blue-radius
75
> (define blue-area(* pi blue-radius blue-radius))
> blue-area
17671.458676442588
> (define red-radius(* base-radius 1/7))
> red-radius
  2
14-
  7
> (define red-area(* pi red-radius red-radius))
> red-area
641.141357875468
> (define outer-red(- base-area blue-area))
> (define red(+ outer-red red-area))
> red
14385.609217330812
> (define percent-red(*(/ red base-area)100))
> percent-red
45.79081632653061
> |
```

Fifth Task: Illustration of the Target Problem



```
> (require 2htdp/image)
> (define radius 100)
> (define base-circle(circle radius "solid" "red"))
> base-circle
> (define blue-radius(* radius 0.75))
> (define blue-circle(circle blue-radius "solid" "blue"))
> blue-circle
> (define red-radius(* radius 1/7))
> (define red-circle(circle red-radius "solid" "red"))
> red-circle
> (define target(overlay red-circle blue-circle base-circle))
> target
> |
```