

3.1 Lesson 1: Getting Acquainted with Racket/DrRacket

*

Learning Abstract- What I learned in this assignment is how to create objects in the Racket programming language as well as computing numbers which are very different from other languages that I have coded in.

3.5 Interactions: Simple Numeric Processing

The following Racket interaction is simply intended to introduce numbers and numeric operators in Racket. What can you say about Racket as a result of the interaction? What questions does the interaction most immediately bring to mind?

```
> 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
> |
```

*

> Interactions: Solution to the Scrap Problem

...

The Scrap Problem: A circular disk of maximal size is cut from a square piece of tin of side 100 units. What is the area of the scrap?

```

> 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
>

```

> Interactions: Illustration of Scrap Problem Situation

```

> ( 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

> |

```

The Target Problem

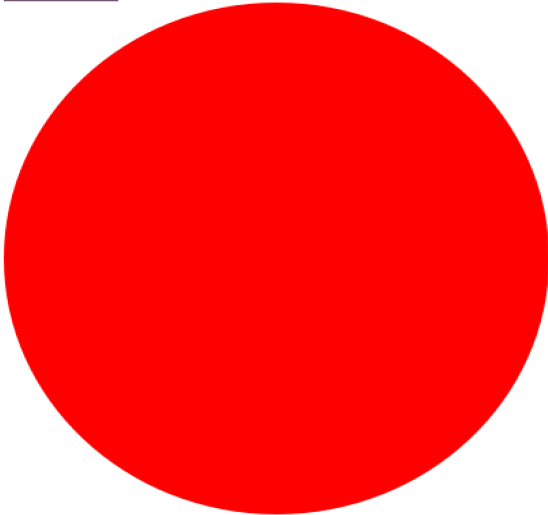
A “target” consists of a red disc of some diameter, containing a blue disc of diameter $\frac{3}{4}$ that of the bigger disc, which, in turn, contains another red disk, this one of diameter $\frac{1}{7}$ that of the biggest disc. To clarify, you should be thinking something like this:

```
> (require 2htdp/image)
```

```
> (define radius 200)
```

```
> (define the-circle (circle radius "solid" "red"))
```

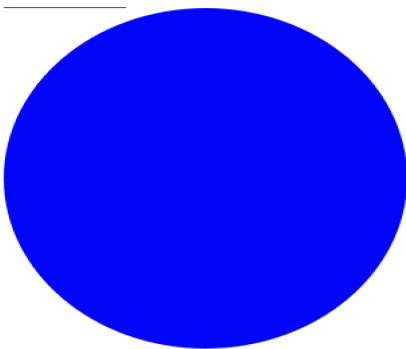
```
> the-circle
```



```
> (define radius2 ( * radius 0.75))
```

```
> (define blue-circle ( circle radius2 "solid" "blue"))
```

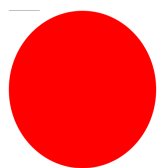
```
> blue-circle
```



```
> (define radius4 ( * radius 0.14285714))
```

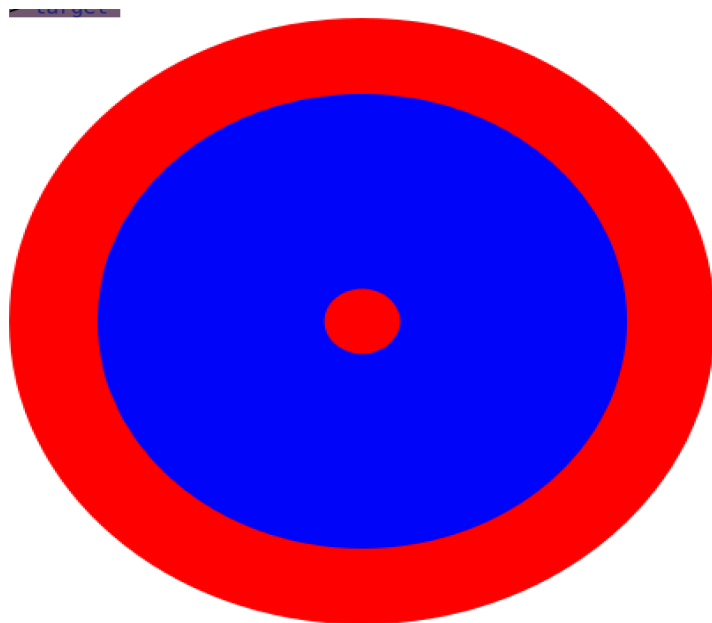
```
> ( define tiny-circle ( circle radius4 "solid" "red"))
```

```
> tiny-circle
```



```
> ( define target ( overlay tiny-circle blue-circle the-circle))
```

```
> target
```



What percentage of the target is red?

16.29%