
First Racket Programming Assignment Solution

Learning Abstract

In this assignment I learned a little bit about numeric computations in Lisp. I also learned how to bind variables to values. I solved a couple of numeric problems by using basic arithmetic operations. And I learned to use a Racket library to create and display shapes so that I could render the problem situations graphically. All of this took place within the Interactions pane of the DrRacket PDE.

Interaction: Simple Numeric 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
> |
```

Interaction: 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
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
>
```

Interaction: 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
```



```
> |
```

Interaction: Illustration of the Target Problem Situation

→the work goes here←

Interaction: Solution to Target Problem

→the work goes here←