Errol Quinones CSC344 – Programming Languages

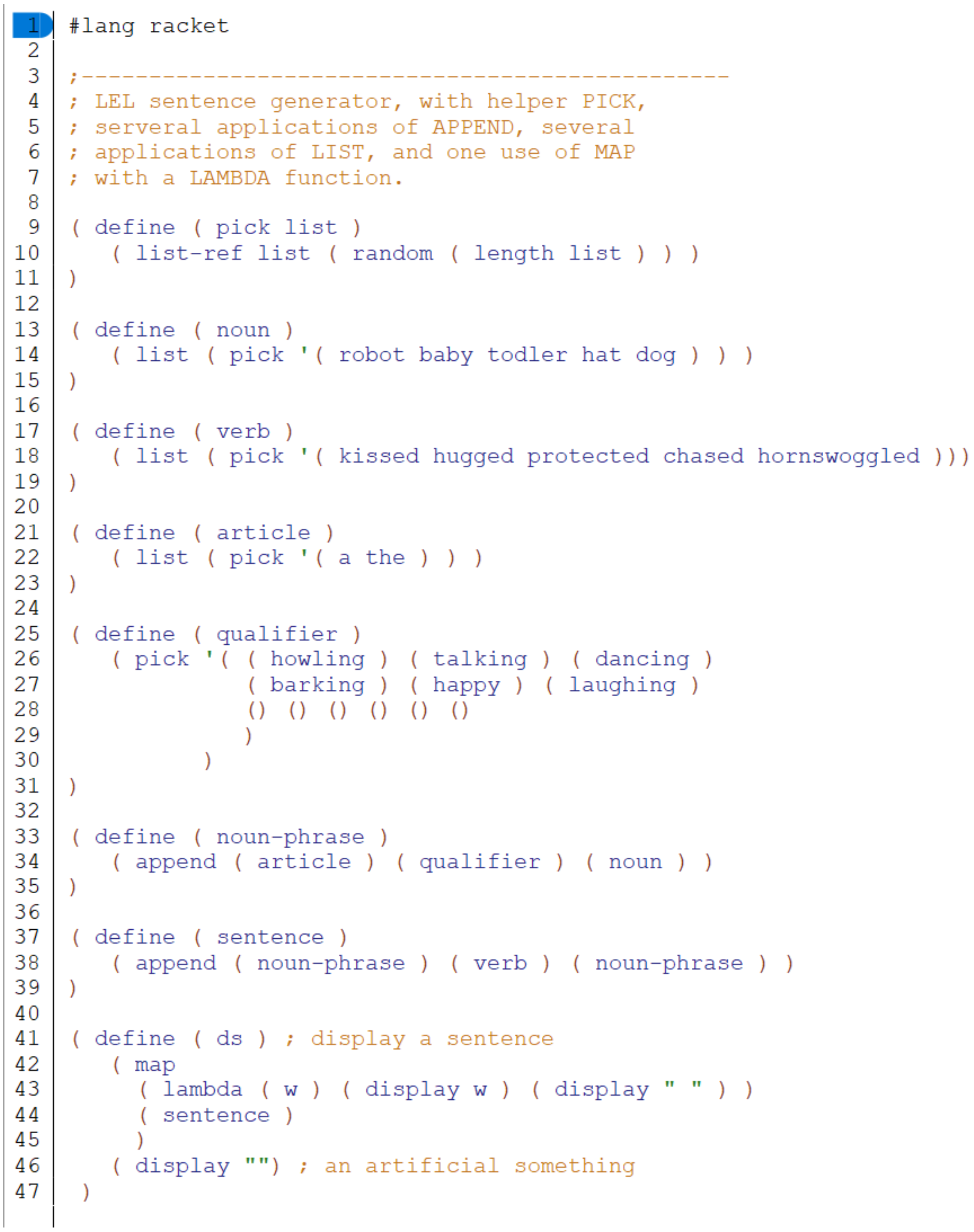
____

Racket Assignment #1: Getting Acquainted with Racket/DrRacket + LEL Sentence Generation.

ABSTRACT:

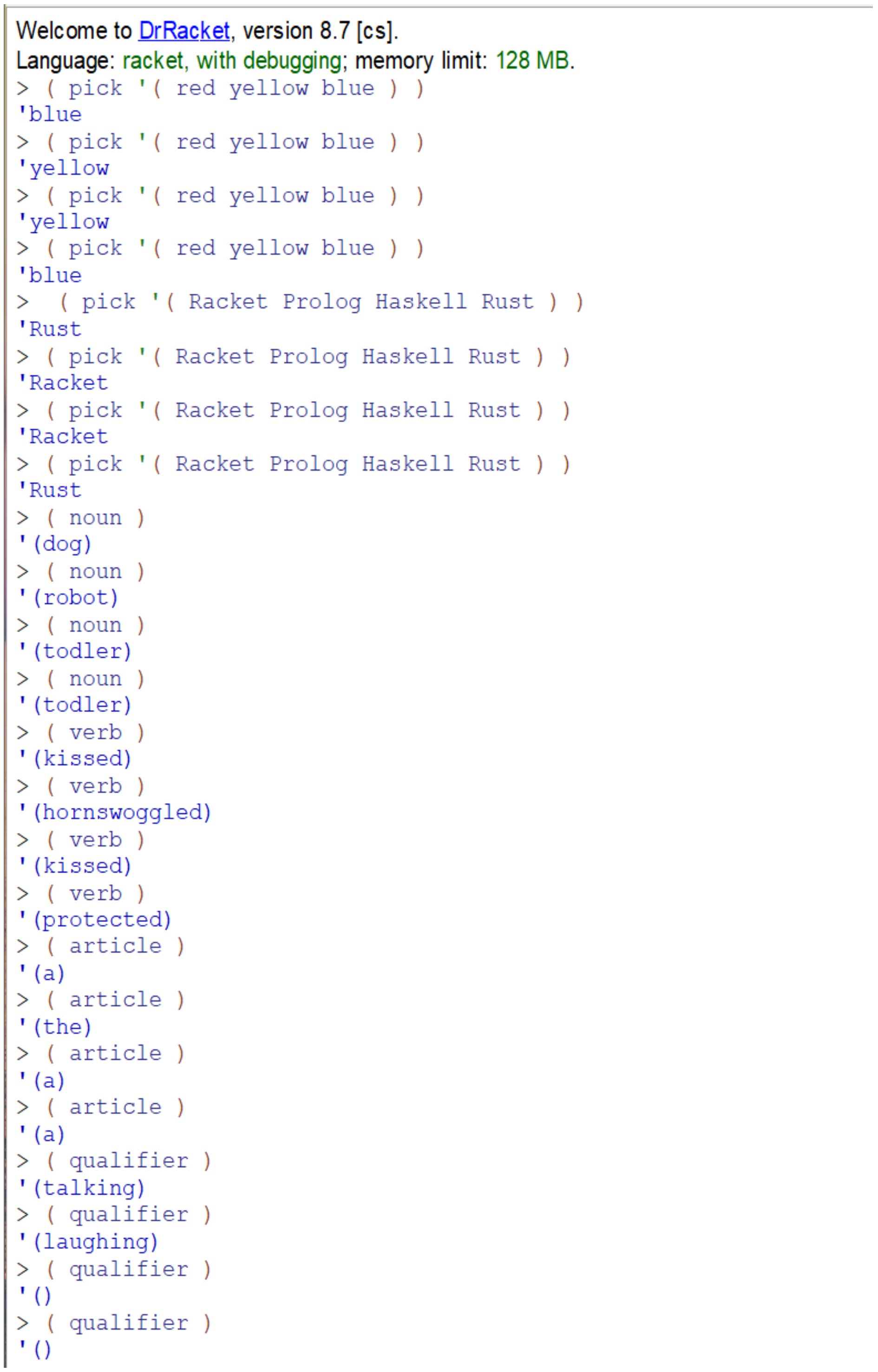
This first programming assignment serves as an introduction to Racket programming in the DrRacket program development environment (PDE). This assignment includes a LEL (Little English Language) sentence generator which is intended to help the user understand the workings of Racket and the functionality of DrRacket. The LEL sentence generator program generates random sentences that follow LEL grammar. Below is the code and a demonstration of the LEL sentence generator.

Code for the LEL Sentence Generator



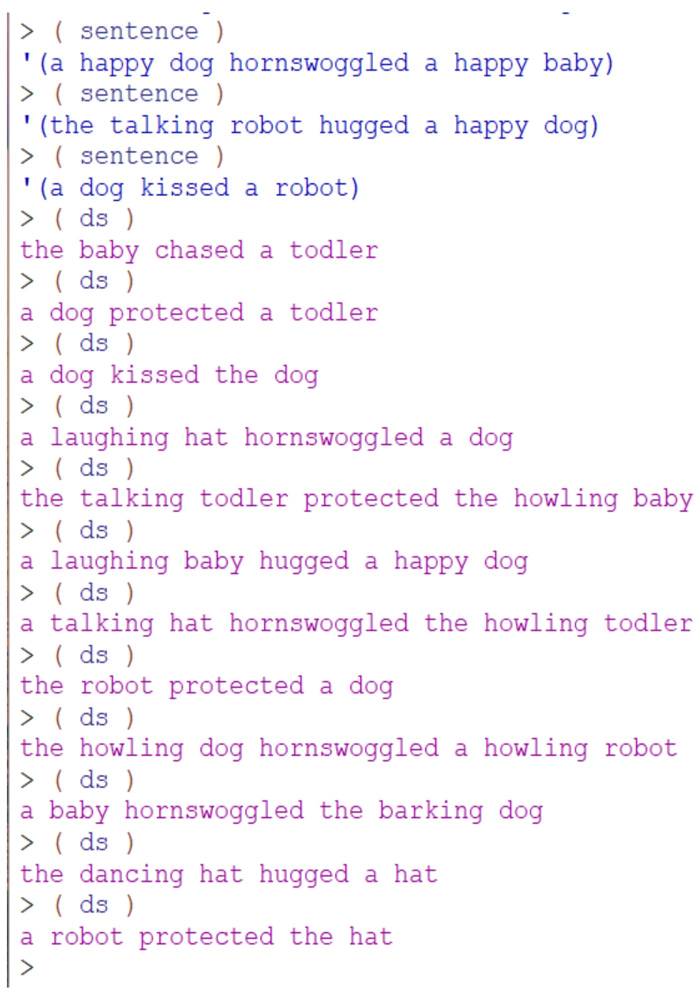
```racket
#lang racket

;-------------------------------------------------
; LEL sentence generator, with helper PICK,
; serveral applications of APPEND, several
; applications of LIST, and one use of MAP
; with a LAMBDA function.

( define ( pick list )
   ( list-ref list ( random ( length list ) ) )
)

( define ( noun )
   ( list ( pick '( robot baby todler hat dog ) ) )
)

( define ( verb )
   ( list ( pick '( kissed hugged protected chased hornswoggled )))
)

( define ( article )
   ( list ( pick '( a the ) ) )
)

( define ( qualifier )
   ( pick '( ( howling ) ( talking ) ( dancing )
             ( barking ) ( happy ) ( laughing )
             () () () () () ()
             )
          )
)

( define ( noun-phrase )
   ( append ( article ) ( qualifier ) ( noun ) )
)

( define ( sentence )
   ( append ( noun-phrase ) ( verb ) ( noun-phrase ) )
)

( define ( ds ) ; display a sentence
   ( map
     ( lambda ( w ) ( display w ) ( display " " ) )
     ( sentence )
     )
   ( display "") ; an artificial something
 )
```

Demo for the LEL Sentence Generator



```
Welcome to DrRacket, version 8.7 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> ( pick '( red yellow blue ) )
'blue
> ( pick '( red yellow blue ) )
'yellow
> ( pick '( red yellow blue ) )
'yellow
> ( pick '( red yellow blue ) )
'blue
>  ( pick '( Racket Prolog Haskell Rust ) )
'Rust
> ( pick '( Racket Prolog Haskell Rust ) )
'Racket
> ( pick '( Racket Prolog Haskell Rust ) )
'Racket
> ( pick '( Racket Prolog Haskell Rust ) )
'Rust
> ( noun )
'(dog)
> ( noun )
'(robot)
> ( noun )
'(todler)
> ( noun )
'(todler)
> ( verb )
'(kissed)
> ( verb )
'(hornswoggled)
> ( verb )
'(kissed)
> ( verb )
'(protected)
> ( article )
'(a)
> ( article )
'(the)
> ( article )
'(a)
> ( article )
'(a)
> ( qualifier )
'(talking)
> ( qualifier )
'(laughing)
> ( qualifier )
'()
> ( qualifier )
'()
```

```
> ( qualifier )
'()
> ( qualifier )
'(happy)
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'(happy)
> ( qualifier )
'(laughing)
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'()
> ( qualifier )
'(happy)
> ( qualifier )
'(dancing)
> ( qualifier )
'(happy)
> ( noun-phrase )
'(a robot)
> ( noun-phrase )
'(a dog)
> ( noun-phrase )
'(a baby)
> ( noun-phrase )
'(a dancing dog)
> ( noun-phrase )
'(the laughing robot)
> ( noun-phrase )
'(the talking dog)
> ( noun-phrase )
'(a talking robot)
> ( noun-phrase )
'(the barking hat)
> ( sentence )
'(a happy hat hugged the dancing hat)
> ( sentence )
'(the happy robot hornswoggled a happy dog)
> ( sentence )
'(the barking robot hugged a dancing dog)
> ( sentence )
'(a dancing todler protected the todler)
> ( sentence )
'(the dancing todler chased a talking todler)
```



```
> ( sentence )
'(a happy dog hornswoggled a happy baby)
> ( sentence )
'(the talking robot hugged a happy dog)
> ( sentence )
'(a dog kissed a robot)
> ( ds )
the baby chased a todler
> ( ds )
a dog protected a todler
> ( ds )
a dog kissed the dog
> ( ds )
a laughing hat hornswoggled a dog
> ( ds )
the talking todler protected the howling baby
> ( ds )
a laughing baby hugged a happy dog
> ( ds )
a talking hat hornswoggled the howling todler
> ( ds )
the robot protected a dog
> ( ds )
the howling dog hornswoggled a howling robot
> ( ds )
a baby hornswoggled the barking dog
> ( ds )
the dancing hat hugged a hat
> ( ds )
a robot protected the hat
>
```