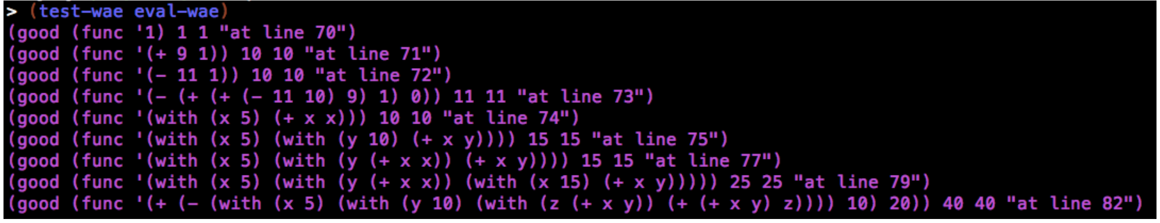
Patrick Walter
2568215
EECS 662
Project 1

Wae Language:

To test the wae language:

1. Open the source code wae_interpretter_parser in DrRacket
2. Run the code (cmd + r)
3. Type: (test-wae eval-wae)
4. Review the output

Sample Output:



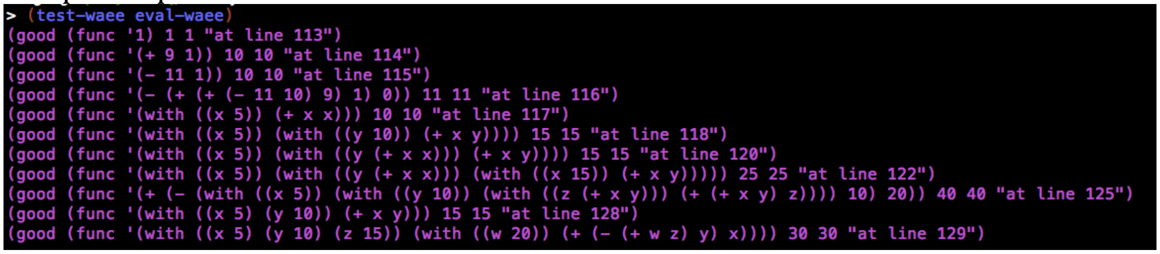
```
> (test-wae eval-wae)
(good (func '1) 1 1 "at line 70")
(good (func '(+ 9 1)) 10 10 "at line 71")
(good (func '(- 11 1)) 10 10 "at line 72")
(good (func '(- (+ (+ (- 11 10) 9) 1) 0)) 11 11 "at line 73")
(good (func '(with (x 5) (+ x x))) 10 10 "at line 74")
(good (func '(with (x 5) (with (y 10) (+ x y)))) 15 15 "at line 75")
(good (func '(with (x 5) (with (y (+ x x)) (+ x y)))) 15 15 "at line 77")
(good (func '(with (x 5) (with (y (+ x x)) (with (x 15) (+ x y))))) 25 25 "at line 79")
(good (func '(+ (- (with (x 5) (with (y 10) (with (z (+ x y)) (+ (+ x y) z)))) 10) 20)) 40 40 "at line 82")
```

Waee language:

To test the waee language

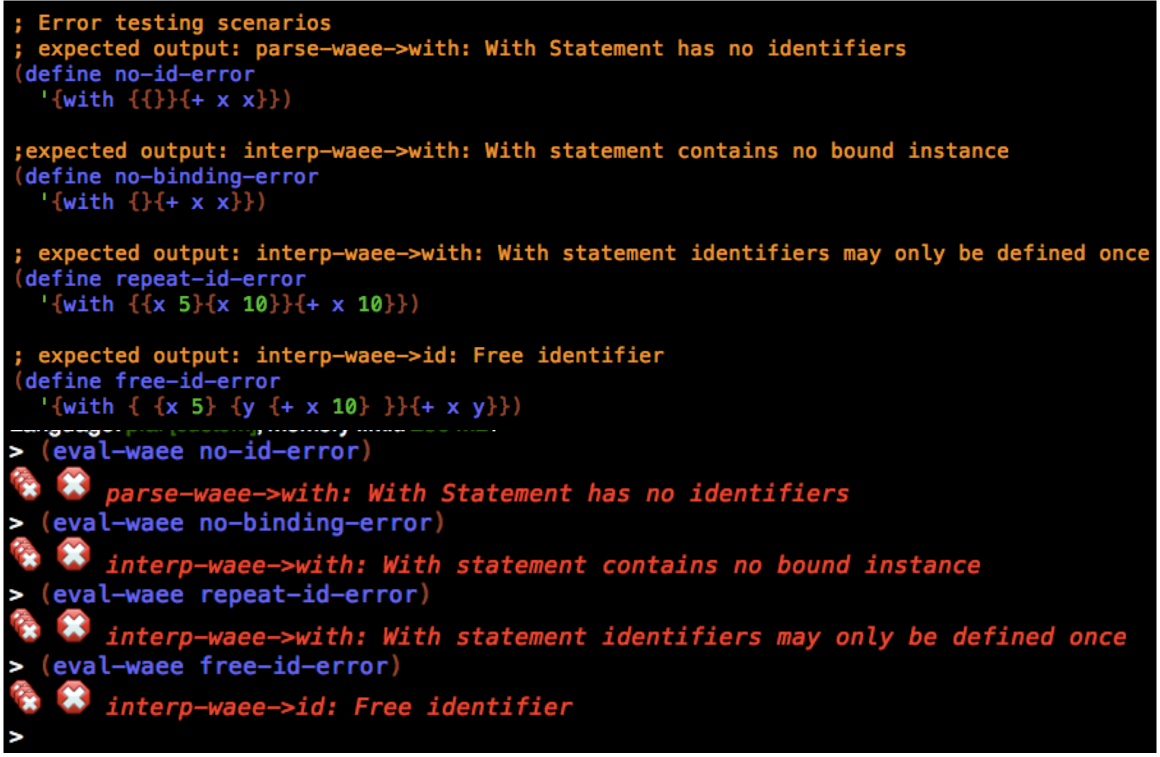
1. Open the source code waee_interpretter_parser in DrRacket
2. Run the code (cmd + r)
3. Type: (test-waee eval-waee)
4. Review the output

Sample output:



```
> (test-waee eval-waee)
(good (func '1) 1 1 "at line 113")
(good (func '(+ 9 1)) 10 10 "at line 114")
(good (func '(- 11 1)) 10 10 "at line 115")
(good (func '(- (+ (+ (- 11 10) 9) 1) 0)) 11 11 "at line 116")
(good (func '(with ((x 5)) (+ x x))) 10 10 "at line 117")
(good (func '(with ((x 5)) (with ((y 10)) (+ x y)))) 15 15 "at line 118")
(good (func '(with ((x 5)) (with ((y (+ x x))) (+ x y)))) 15 15 "at line 120")
(good (func '(with ((x 5)) (with ((y (+ x x))) (with ((x 15)) (+ x y))))) 25 25 "at line 122")
(good (func '(+ (- (with ((x 5)) (with ((y 10)) (with ((z (+ x y))) (+ (+ x y) z)))) 10) 20)) 40 40 "at line 125")
(good (func '(with ((x 5) (y 10)) (+ x y))) 15 15 "at line 128")
(good (func '(with ((x 5) (y 10) (z 15)) (with ((w 20)) (+ (- (+ w z) y) x)))) 30 30 "at line 129")
```

Sample error output:



```
; Error testing scenarios
; expected output: parse-waee->with: With Statement has no identifiers
(define no-id-error
  '{with {{}}{+ x x}})

;expected output: interp-waee->with: With statement contains no bound instance
(define no-binding-error
  '{with {}{+ x x}})

; expected output: interp-waee->with: With statement identifiers may only be defined once
(define repeat-id-error
  '{with {{x 5}{x 10}}{+ x 10}})

; expected output: interp-waee->id: Free identifier
(define free-id-error
  '{with { {x 5} {y {+ x 10} }}{+ x y}})
```

```
> (eval-waee no-id-error)
  parse-waee->with: With Statement has no identifiers
> (eval-waee no-binding-error)
  interp-waee->with: With statement contains no bound instance
> (eval-waee repeat-id-error)
  interp-waee->with: With statement identifiers may only be defined once
> (eval-waee free-id-error)
  interp-waee->id: Free identifier
>
```