SYED MUHAMMAD SHAHZEB NAQVI

P20101036

PROGRAMMING LANGUAGES ADVANCED

LAB-01

MCS(PREVIOUS)

EXERCISE_01: In exercise one you will learn how to execute basic mathematical functions in the racket

language.

A) 23+5

Ans: (+23 5)

> 28

B) 15-6

Ans: (- 15 6)

> 9

C) 24/6

Ans: (/ 24 6)

> 4

D) 5*9

Ans: (* 5 9)

> 45

```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (+ 23 5)
28
> (- 15 6)
9
> (/ 24 6)
4
> (* 5 9)
45
```

EXERCISE_02: Use what you have learned about the basic operations to now complete these order of

operations problems using racket.

A) 3*5+2-3

Ans: (- (+ (* 3 5) 2) 3)

> 14

B) 20 – 5 * 3

Ans: (- (* 5 3) 20)

> -5

C) 12 / 2 +5 – 3 * 2

Ans: (- (+ (/ 12 2) 5) (* 3 2))

> 5

```
> (- (+ (* 3 5) 2) 3)
14
> (- (* 5 3) 20)
-5
> (- (+ (/ 12 2) 5) (* 3 2))
5
>
```

Determine language from source ▾ 17:0 P 508.72 MB ()

EXERCISE_03: Writing functions

Write your own function called "squareit" that will square a number. (Multiply a number times itself). Test out your function with 3 different numbers.

```
> (define (squareit x) (* x x))
> (squareit 2)
4
> (squareit 3)
9
> (squareit 4)
16
```

EXERCISE_04: Next, write a function called "cubeit" that will cube a number. (Multiply a number times itself 3 times.... 2^3= 8 because 2*2=4*2=8

```
> (define (cubeit x) (* x x x))
> (cubeit 4)
64
> (cubeit 9)
729
> (cubeit 11)
1331
>
```

EXERCISE_05: Write a function called "double" that will double a number. (There are two different correct ways to write this one! Can you think of both?)

1st Way is to add a number to itself.

```
> (define (doubleadd x) (+ x x))
> (doubleadd 9)
18
> (doubleadd 99)
198
>
```

2nd Way is to multiply a number by 2.

```
> (define (doublemult x) (* x 2 ))
> (doublemult 5)
10
> (doublemult 35)
70
>
```

EXERCISE_06: Write a function called “upplace” that will take number and increase the place value. For example (upplace 3) => 30 or (upplace 44) => 440.

```
> (define (upplace x)(* x 10))
> (upplace 99)
990
> (upplace 323)
3230
>
```

```
Welcome to DrRacket, version 8.2 [cs].
Language: racket, with debugging; memory limit: 128 MB.
> (+ 23 5)
28
> (- 15 6)
9
> (/ 24 6)
4
> (* 5 9)
45
> (- (+ (* 3 5) 2) 3)
14
> (- (* 5 3) 20)
-5
> (- (+ (/ 12 2) 5) (* 3 2))
5
> (define (squareit x) (* x x))
> squareit 2
#<procedure:squareit>
2
> (define (squareit x) (* x x))
> (squareit 2)
4
> (squareit 3)
9
> (squareit 4)
16
> (define (cubeit x) (* x x x))
> (cubeit 4)
64
> (cubeit 9)
729
> (cubeit 11)
1331
> (define (doubleadd x) (+ x x))
> (doubleadd 9)
18
> (doubleadd 99)
198
> (define (doublemult x) (* x 2 ))
> (doublemult 5)
10
> (doublemult 35)
70
> (upplace 2)
. . upplace: undefined;
 cannot reference an identifier before its definition
> (define (upplace x)(* x 10))
> (upplace 99)
990
> (upplace 323)
```