Warm-Up Quiz

Bookmark this page

Warm-Up 05.1

1/1 point (graded)

According to the introduction, which important programming technique will be introduced in this module?

- ○ Eversion
- ○ Perversion
- ● Recursion
- ○ Regression
- ○ Subversion

✓

Save Show answer

Submit You have used 1 of 2 attempts

Warm-Up 05.2

1/1 point (graded)

When testing as part of the Design Recipe, different functions will require different numbers of tests. Which statement about the number of tests required is accurate?

- ○ Every function should have exactly five tests
- ○ The number of tests should be equal to the number of lines of code in the function
- ○ The number of tests should be equal to the number of function parameters
- ○ The number of tests should be equal to the number of characters in the function's name
- ● None of the above

✓

Save Show answer

Submit You have used 1 of 2 attempts

Warm-Up 05.3

1/1 point (graded)

There are important differences between strings and symbols in Racket. Which of the following is *not* a valid difference between the two types?

- ○ Strings are enclosed by double quotes (`"`), symbols are preceded by single quotes (`'`)
- ○ A symbol is considered atomic, whereas a string contains data that can be extracted and processed
- ● We can define a constant whose value is a string, but not one whose value is a symbol
- ○ Strings are well suited for storing answers to open-ended "fill in the blank" questions, whereas symbols are well suited for limited "multiple choice" questions
- ○ We can determine how long a string is using the built-in function `string-length` ; no equivalent function exists for symbols

✓

Save Show answer

Submit You have used 1 of 2 attempts

Warm-Up 05.4

1/1 point (graded)

What is the value of `s4` after executing the code below?

```
(define s1 "Waterloo")
(define s2 (string-append s1 s1))
(define s3 (substring s2 3 6))
(define s4 (substring s3 1 (sub1 (string-length s3))))
```

- ○ `"t"`
- ● `"r"`
- ○ `"ter"`
- ○ `"rlo"`
- ○ No value: there's an error in the code

✓

Save Show answer

Submit You have used 1 of 2 attempts

Warm-Up 05.5

1/1 point (graded)

Assume that the following definition has been completely evaluated:

```
(define a 5)
```

Now consider the following Racket expression:

```
(and
  (cond [false true] [false 6])
  (< a 17))
```

Which substitution rule will be applied next in a trace of the expression? You don't need to apply the rule or know the result; just decide which rule would be used next.

- ○ *small-sub*
- ○ *big-sub*
- ○ *as-if-by-magic*
- ○ `(and false exp ...) ⇒ false`
- ● `(cond [false ...] [exp1 exp2] ...) ⇒ (cond [exp1 exp2] ...)`

✓

Save Show answer

Submit You have used 1 of 2 attempts

Warm-Up 05.6

1/1 point (graded)

Let's see if we can use the substitution rules *backwards in time*! Consider this short Racket program.

```
(define a 12)
(or false (> a 7))
```

Of the programs shown below, which one *could not* have been the step immediately before the one above in a trace using the standard substitution rules?

○
```
(define a (* 3 4))
(or false (> a 7))
```

○
```
(define a 12)
(or false false (> a 7))
```

●
```
(define a 12)
(or (odd? a) (> a 7))
```

○
```
(define a 12)
(or (and false (cond [(even? a) false] [else true])) (> a 7))
```

○
```
(define a 12)
(cond
  [true (or false (> a 7))]
  [else (or false (< a 7))])
```

✓

Save Show answer

Submit You have used 1 of 2 attempts

Warm-Up 05.7

1/1.0 points
Attempts: 1 / Unlimited

Assume that the following definitions have been fully evaluated:

```
(define x (- (sqrt 36)))
(define y (* (+ x 10) 2))
(define (f x y)
  (+ 9 (* 2 x y) (- x y)))
```

Complete the trace of the following expression.

```
  (f y x)
⇒ (f 8 x)
⇒ (f 8 -6)
⇒ (+ 9 (* 2 8 -6) (- 8 -6))
⇒ (+ 9 -96 (- 8 -6))
⇒ (+ 9 -96 14)
⇒ -73
```

Submit Step Final Value Stop With Error Reset Step

✓ Correct! 1/1.0 points

Warm-Up 05.8

Newton's method tells us that for a given n, if x is an approximation of $\sqrt{n}$, then

$$\frac{1}{2}\left(x + \frac{n}{x}\right)$$

is an even better approximation of $\sqrt{n}$. Write a Racket function `newton` that consumes an integer n (which may be negative) and a number x and computes the next Newton approximation according to the formula above.

1/1 points
Attempts: 1 / Unlimited

```
(define (newton n x)
  (* (/ 1 2) (+ x (/ n x))))
```

Submit Code Run Code Reset Code

✓ Correct! 1/1 points

Warm-Up 05.9

1/1 point (graded)

In the space below, give the correct contract for the function `newton` in the previous question. Be sure to use the correct spelling, spacing, and punctuation, and remember to write the contract as a comment. You only need to give the first line of the contract; ignore any Requires clause that may be needed.

;; newton: Int Num -> Num ✓

Save Show answer

Submit You have used 1 of 5 attempts

Warm-Up 05.10

Write a function `starts-with?` that consumes two strings, `str1` and `str2`. The function produces `true` if and only if `str1` starts with `str2`. Thus `(starts-with? "Pizza" "Pi")` would produce `true`, and `(starts-with? "Calzone" "Dal")` would produce `false`. Note that you must check that `str1` is at least as long as `str2`, and remember that every string starts with the empty string (`""`).

1/1 points
Attempts: 1 / Unlimited

```
;; starts-with?: Str Str -> Bool
(define (starts-with? str1 str2)
  (cond
    [(and (>= (string-length str1) (string-length str2))
  (string=? (substring str1 0 (string-length str2)) str2)) true]
[else false]))
```

Submit Code Run Code Reset Code

✓ Correct! 1/1 points

Discussion

Topic: Module 05 / Warm-Up Quiz

Hide Discussion

Add a Post

Show all posts by recent activity

? 5.7
I am stuck on the first step of the trace and am unsure what I am messing up on. I have come up with : (+ 9 (* 2 8 -6) (- 8 -6))) for the first step of the trace. Any help would be ... 2

? Module 5 due date
Hi, the due date for module 5 on this site says Feb 11 but on learn it is Feb 4, so I'd just like to confirm which is the correct one? Also what is the date for the 5% midterm quiz, o... 2