

Some Racket/C++ Comparisons

Comments

Racket

```
; single line comment

#| multi
  line
  comment |#
```

C++

```
// single line comment

/* multi
  line
  comment */
```

Data types

Racket

```
boolean -> #true, #false

number -> 3, -3, 3.0, 3e5, 3/5

string -> "How are you?", "a"
```

C++

```
bool -> true, false

int -> 3, -46, +568

double -> 3.0, 3e5

char* -> "How are you?", "a"

// BUT! C++ has a string class we will be
//      using as our first choice to represent
//      "stringy" data.

// YOU CAN USE a char* argument for a string
//      parameter!

char -> 'a', 'G', '\n'
```

Named Constants

Racket

```
(define MAX-RED-DESIRED 200)

(define MY-PI 3.14159)

(define LANG "Racket")

(define START-FLAG #true)
```

C++

```
const int MAX_RED_DESIRED = 200;

const double MY_PI = 3.14159;

const string LANG = "C++";

const bool START_FLAG = true;
```

Functions

Racket

```
; signature: rect-area: number number -> number
; purpose: expects a rectangle's length and width, and returns
;         that rectangle's area

(define (rect-area length width)
  (* length width)
)

(check-expect (rect-area 3 5) 15)
(check-within (rect-area 4.6 6) (* 4.6 6) 0.001)

(rect-area 4.6 6)
```

C++

```
/*---
  signature: rect_area: double double -> double
  purpose: expects a rectangle's length and width, and returns
           that rectangle's area
  tests:
    rect_area(3, 5) == 15
    abs( rect_area(4.6, 6) - (4.6 * 6) ) < 0.001

  compile in CS50 IDE terminal (open to same folder as this file) using:
    g++ test.cpp -o test
  run in CS50 IDE terminal (open to same folder as this file) using:
    ./test
---*/

#include <cstdlib>
#include <iostream>
#include <string>
#include <cmath>
using namespace std;

double rect_area(double length, double width)
{
  return length * width;
}

/* testing rect_area */

int main()
{
  cout << boolalpha;
  cout << "***** Testing rect_area *****" << endl;

  cout << (rect_area(3, 5) == 15) << endl;
  cout << ( abs( rect_area(4.6, 6) - (4.6 * 6) ) < 0.001 ) << endl;

  cout << rect_area(4.6, 6) << endl;

  return EXIT_SUCCESS;
}
```