HW0 – Jelle Piepenbrock – 11:44 Amsterdam time (GMT +1) Nov 02

I found this assignment to be a good refresher of what I already knew about Ruby Programming. At first, I was sort of stumped because I seemed to have forgotten a great deal since my last attempt to learn Ruby, but after the first few exercises, it all came back. I did not do pair programming, or a video recording because I did these exercises in my university library, on the public access computers, because my own computer is unfortunately not working correctly. I hope to have it fixed by next week, so I can do a pair programming assignment. They seem to be very helpful.

I googled around a lot when I did not know how to solve a problem, and I also stumbled upon some ways to shorten my answers. For example, some of my if/else constructions could in principle be changed into single-line statements using somewhat more advanced Ruby code, but I chose to keep it like I came up with them, so at least I fully understood what was happening (though I will definitely try to use those exotic constructions in the next assignment!)

The last thing that I want to say is actually something I was worried about, because the autograder only gave me 25/100 as a score, even though it gives the result below. I passed all the tests, 308 out of 308, and submitted it before the deadline (which was 23:59 on 2015-11-02). So something is wrong there: even if I was late, which I believe I was not, I should still get a 75/100 score. I also noticed that it said that the files took more than 500 minutes to load, which is just nonsense I think.)

Your submission was recorded at 2015-11-02 10:26:00 +0000 : submission is late and scaled by 0.25

Score: 308/308

BookInStock

should be defined

getters and setters

should set ISBN [10 points]

should set price [10 points]

should be able to change ISBN [10 points]

should be able to change price [10 points]

constructor

should reject invalid ISBN number [10 points]

should reject zero price [10 points]

should reject negative price [10 points]

#price\_as\_string

should be defined

should display 33.95 as "$33.95" [10 points]

should display 1.1 as $1.10 [10 points]

should display 20 as $20.00 [10 points]

Finished in 0.00805 seconds (files took 544 minutes 6 seconds to load)

12 examples, 0 failures

Ruby intro part 1

#sum

should be defined

returns correct sum [20 points]

works on the empty array [10 points]

#max\_2\_sum

should be defined

returns the correct sum [7 points]

works even if 2 largest values are the same [3 points]

returns zero if array is empty [10 points]

returns value of the element if just one element [10 points]

#sum\_to\_n

should be defined

returns true when any two elements sum to the second argument [30 points]

returns false for the single element array [5 points]

returns false for the empty array [5 points]

Finished in 0.00774 seconds (files took 544 minutes 6 seconds to load)

12 examples, 0 failures

#hello

should be defined

The hello method returns the correct string [30 points]

#starts\_with\_consonant?

should be defined

classifies true cases [10 points]

classifies false cases [10 points]

works on the empty string [5 points]

works on nonletters [5 points]

#binary\_multiple\_of\_4?

should be defined

classifies valid binary numbers [30 points]

rejects invalid binary numbers [10 points]

Finished in 0.00373 seconds (files took 544 minutes 6 seconds to load)

10 examples, 0 failures