

[Question 1 (5 pts)] What is the return value of the following Ruby expressions?

- | | |
|--|-------|
| • <code>[]</code> .class | Array |
| • <code>3 / 2</code> | 1 |
| • <code>3.0 / 2.0</code> | 1.5 |
| • <code>[]</code> .nil? | false |
| • <code>def h; "Hello world"; end</code> | nil |

[Question 2 (5 pts)] What is your personal favorite feature of Ruby? Why?

Ruby's fully object-oriented environment is quite intuitive. By dousing the need for multiple-inheritance, Ruby gets rid-of the confusion associated with handling numerous classes.

[Question 3 (5 pts)] In your opinions, are the advantages of BDD worth the drawbacks (at least in theory)? Why or why not?

Yes, BDD helps both the programmer and client better-comprehend the features to be implemented by the system

[Question 4 (5 pts)] Now that you have used BDD in a development process, do you think it helps or hinders developers?

BDD helps for large-scale/high-risk projects; however, programs similar to Cucumber only hinder programmers when used in-parallel with small-scale/low-risk projects. (i.e. This lab's assignment could have been made in a quarter of the time if Cucumber was not used.)

[Question 5 (5 pts)] What is the advantage of using more verbose, complex English in your feature definitions? What is the drawback?

Verbose language usage allows readers to understand the features faster and with greater ease; however being more complex, the definitions take much longer to write and are also limited due to increased specificity.