

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

```
[] .class  
3 / 2  
3.0 / 2.0  
[].nil?  
def h; "Hello world"; end  
  
.class = syntax error  
Something.class = name error  
  
3/2=1  
3.0/2.0=1.5  
Nil? =false  
  
def h; "Hello world"; end  
=nil but if you input h you get Hello world
```

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

The simplicity of the statements and how you don't have to initialize the variables

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

In my opinion, yes because you are making larger steps forward and you are customer centered. The drawbacks are just like the TTD so I don't see the difference. They both can give false positives and they both have you write tests for code you haven't written yet.

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

Well I think it hinders, it is way easier to just write unit tests but it is a lot easier to read

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

It makes the tests easier to read but the drawbacks are it takes longer to write