

Lab 9 Questions

1.

```
> [].class  
=> Array  
  
> 3 / 2  
=> 1  
  
> 3.0 / 2.0  
=> 1.5  
  
> [].nil?  
=> false  
  
> def h; "Hello world"; end  
=> nil
```
2. I enjoy the way that we can define a class with methods and all by sequential prompt lines. I also like the fact that everything is a class.
3. I believe BDD would be worth the drawbacks given the project is a long-term consumer based product. Creating a DSL may mean a lot of extra work, but with BDD we start with an overall goal, which gives us much more direction in our development as opposed to TDD.
4. I believe that BDD helps developers by ensuring they are building a project exactly to specification. Once a feature definition is written, we can write code to suit it, eliminating extraneous code. BDD could be called a hindrance to developers due to the extra time it takes to write English and Ruby code, but BDD is a safe tactic overall because it gears the product to customer wants and needs.
5. The more verbose we are with our feature definition, the easier it is for us and others to understand exactly what the program is supposed to do. Writing detailed scenarios helps us write our code to handle specific situations. The drawback to complex definitions is the fact that everything we write must be mapped to code.