TP#6: February 09th – Design Patterns, Refactoring, Code Smells

What? How can code "smell"??? Well, it doesn't have a nose... but it definitely can stink!

I cannot stress enough how important and well written the refactoring.guru website is. It truly is a golden mine of good practices, incredibly well documented code smells, masterfully executed design patterns and more. Deadline is Sunday 13th. Question's answer should not be in the readme.

Exercise 1:

Scroll through the refactoring part of the website. With your newfound knowledge, you should easily find multiple code smells in the "Guilded Rose" Kata and be able to answer these questions:

1. What is clean code? What is the role played by refactoring one's code?
2. Do you think you can "over-refactor" and do too much? How?
3. What is a code smell? Why should you bother?
4. Can you identify a few code smells in the original GuildedRose?
5. What are some refactoring techniques you could have used in the GuildedRose?

Exercise 2:

Now get to the other part of this website, under "Design Patterns". You'll discover incredible mechanism used to build robust software that are used through-out the world.

1. In simple terms, and a few sentences at most, what's a design pattern?
    a. When should you use one?
    b. When shouldn't you?
2. Write a small program that build a pizza with various toppings using the builder pattern. Why is this a good idea? Think about scalability for instance
3. Find an original idea to implement a decorator pattern (not the one from the website). Can you think of any limitations? No need to code here.
4. Write a small program to implement the behavioral design pattern of your choice. Once again, do not use the example given by the website, but feel free to use them as guidance.

API Call (not an exercise):

If you're fast enough and that most of the Exercise is done, you can come to me and together we'll give a look at the CryptoCompare API, what the parameters in the url are like, how to make a simple API call, how to properly deserialize JSON into proper object, and maybe even how to work with the data. You're expected to have signed up to the API and have a proper API key by this point.