Single Responsibility Principle by Sean Chambers

After Chad and Ray, I followed suit as well and am doing Pablo's Topic of the month post on the Single Responsibility Principle or SRP for short.

In SRP a reason to change is defined as a responsibility, therefore SRP states, "An object should have only one reason to change". If an object has more than one reason to change then it has more than one responsibility and is in violation of SRP. An object should have one and only one reason to change.

Let's look at an example. In the example below I have a BankAccount class that has a couple of methods:

```
public abstract class BankAccount
{
    double Balance { get; }
    void Deposit(double amount) {}
    void Withdraw(double amount) {}
    void AddInterest(double amount) {}
    void Transfer(double amount, IBankAccount toAccount) {}
}
```

Let's say that we use this BankAccount class for a person's Checking and Savings account. That would cause this class to have more than two reasons to change. This is because Checking accounts do not have interest added to them and only Savings accounts have interest added to them on a monthly basis or however the bank calculates it.

Some people may say that the class would even have 3 reasons to change because of the Deposit/Withdraw methods as well but I think you can definitely get a little crazy with SRP. That being said, I believe it just depends on the context.

So, let's refactor this to be more SRP friendly.

```
public abstract class BankAccount
{
    double Balance { get; }
    void Deposit(double amount);
    void Withdraw(double amount);
    void Transfer(double amount, IBankAccount toAccount);
}

public class CheckingAccount : BankAccount
{
}
```

```

public class SavingsAccount : BankAccount
{
   public void AddInterest(double amount);
}
```

So what we have done is simply create an abstract class out of BankAccount and then created a concrete CheckingAccount and SavingsAccount class so that we can isolate the methods that are causing more than one reason to change.

When you actually think about it, every single class in the .Net Framework is violating SRP all of the time. The GetHashCode() and ToString() methods are causing more than one reason to change, although you could say that these methods are exempt because they exist in the framework itself and out of our reach for change.

I'm sure you can come up with a lot more instances where you have violated SRP, and even instances where it just depends on the context. As stated on Object Mentor: "The SRP is one of the simplest of the principle, and one of the hardest to get right".

Here is a link to the SRP pdf on Object Mentor for more information.