Options Pattern

We often have settings set in separate configuration files as part of our applications. These can be susceptible to errors due to often needing to request each setting value direct from the file and often by using hard coded strings to match the setting/key names.

Instead of doing this we can use the Options Pattern where we create a C# class object to represent the settings which has Properties with getters and setters to match the desired configuration options. With this we can retrieve our options form the file once and store it in our object. Everywhere we need the settings values we retrieve them from the object, instead of re-reading the file.

This gives us better error handling because the only place the strings are used will be in the object for the initial retrieval of the setting. Additionally, when we are using the property names through the rest of our code, IDE will inform us if we are using it wrong or spell it incorrectly, unlike if we were using the value names inside strings.

Settings File

```
{
    “key”:{
            “config1”:”value”,
            “config2”:”value”,
    }
}
```

Settings Class

```
Public class Key
{
    public string Config1{ get;  set; }
    public string Config2{ get;  set; }
}
```