How to do TDD?

- We have a user story, use 3 step rule to make your idea (baby requirement)

  Allow user input a string to set the display name when he login. Max length of string is 125 characters
How to do TDD?

- String can contain special characters?
- What kind of special characters (!@#$%^&…)?
- String can contain space?
- String can contain number?
- Min length of string?
- String can not contain select, delete, update because of SQL injection
- String can be duplicate?

**Note:** Don’t think the requirement is very simple therefore we don’t need to confirm with Product Owner. Confirmation is a way to prevent bugs.
Example 1: write a method that reverse last 2 characters of string.
If null -> return null, if empty -> return empty, if length of string equal 1 -> return itself
TDD by examples

- Create first test case

```java
[TestClass]
0 references
public class StringHelperTest
{
    [TestMethod]
    0 references
    public void ReverseSuccess_WhenInputHas2Characters()
    {
        StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
    }
}
```
TDD by examples

- Can not compile because StringHelper class is not created
TDD by examples

- After StringHelper class is created, run ALL TEST CASE to see they (or one of them) fail.
TDD by examples

- Make **a little change** to pass this test cases

```csharp
public class StringHelper
{
    public string SwapLast2Chars(string input)
    {
        string first = input[0].ToString();
        string second = input[1].ToString();
        return (second + first).ToString();
    }
}
```

**Question**: Who has a better implementation?
TDD by examples

- Run ALL TEST CASES to see they pass
TDD by examples

- We have just finished a baby step of TDD cycle. Repeat it again.
TDD by examples

- After StringHelper class is created, run **ALL TEST CASE** to see they (or one of them) fail.
TDD by examples

- Refactor code to pass this test case

```csharp
public string SwapLast2Chars(string input)
{
    int length = input.Length;
    string stringMinus2LastChars = input.Substring(0, length - 2);
    string secondLastChar = input[length - 2].ToString();
    string lastChar = input[length - 1].ToString();
    return stringMinus2LastChars + lastChar + secondLastChar;
}
```
TDD by examples

- Run all test cases to see they pass
TDD by examples

- Refactor test code if necessary

```java
public class StringHelperTest {

    StringHelper helper = new StringHelper();

    [TestMethod]
    public void ReverseSuccess_WhenInputHas2Characters() {
        //StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
    }

    [TestMethod]
    public void ReverseSuccess_WhenInputHas4Characters() {
        //StringHelper helper = new StringHelper();
        Assert.AreEqual(helper.SwapLast2Chars("ABCD"), "ABDC");
    }
}
```
TDD by examples

- Add new test case

```java
StringHelper helper = new StringHelper();

[TestMethod]
public void ReverseSuccess_WhenInputHas2Characters()
{
    Assert.AreEqual(helper.SwapLast2Chars("BA"), "AB");
}

[TestMethod]
public void ReverseSuccess_WhenInputHas4Characters()
{
    Assert.AreEqual(helper.SwapLast2Chars("ABCD"), "ABDC");
}

[TestMethod]
public void ReverseSuccess_WhenInputHas1Characters()
{
    Assert.AreEqual(helper.SwapLast2Chars("A"), "A");
}
```
TDD by examples

- After StringHelper class is created, run **ALL TEST CASE** to see they (or one of them) fail
Refractor code to pass the test case
TDD by examples

- Run all test cases to see they pass
TDD by examples

- Add new test case, this test case will be pass without any changes
TDD by examples

- Add new test case, this test case will be fail

```csharp
[TestMethod]
public void ReverseSuccess_WhenInputIsNonNull()
{
    Assert.AreEqual(helper.SwapLast2Chars(null), null);
}
```

Failed Tests (1)
- ReverseSuccess_WhenInputIsNonull 69 ms
TDD by examples

- Refactor code

```csharp
public string SwapLast2Chars(string input)
{
    if (input == null)
    {
        return null;
    }
    int length = input.Length;
    if (length < 2)
    {
        return input;
    }
    string stringMinus2LastChars = input.Substring(0, length - 2);
    string secondLastChar = input[length - 2].ToString();
    string lastChar = input[length - 1].ToString();
    return stringMinus2LastChars + lastChar + secondLastChar;
}
```
TDD by examples

- Run all test cases, if they pass and you don’t have any test case. The TDD cycle is completed. Refactor code if necessary