Code-Reading interview question:

This function is used to display a welcome message.  At first there was just the need for one, but new versions require a second.  For backwards compatibility the old message is still returned under the correct conditions.  Some special cases were needed for specific requests.

It is also assumed a message is always returned no matter what the input.

This program is not written in any particular language (though it resembles C# and Java). The syntax should be taken to mean what you intuit it to mean. Any tests that you feel exercise interpretation of operations can be specially noted in your test-case title.

Three constants are used in this fragment.  For the purposes of this question, assume that:

OLD\_VALUE = "Hello OLD"

NEW\_VALUE = "Hello NEW"

VERSION\_THRESHOLD = 3

Write various testcases that demonstrate the various output possible by the function, and what type of scenario this test covers (as in, if this test was automated, what would its description be). See far below for example.

This function is not written in any particular language, treat it how you would interpret it. If testcases fall out of that interpretation, state your assumptions.

private String getWelcomeMessage(int PlatformID, String PlatformType, int Appversion)

{

     if (PlatformType == "SpecialDevice" && AppVersion == 1 && (PlatformID == 1 || PlatformID == 2) )

     {

           return OLD\_VALUE;

     }

     bool shownew = false;

     switch (PlatformID)

     {

           case 1:

                shownew = true;

                break;

           case 2:

                shownew = true;

           case 3:

                return OLD\_VALUE;

           case 4:

                shownew = true;

                break;

     }

     if (shownew) {

           if (AppVersion < VERSION\_THRESHOLD) {

                return OLD\_VALUE;

           }

           else {

                return NEW\_VALUE;;

           }

     }

     return null;

}

**Test Case Listing:**

Add your testcases below

**Template:**

Input:

PlatformID =

AppVersion =

PlatformType =

Expected Result:

Reason:

***Example*** *(may not be necessarily a correct example):*

Input:

PlatformID = 1

AppVersion = 10

PlatformType = anything

Expected Result:

"Hello NEW"

Reason:

Nominal case for arbitrarily high AppVersion > 3

**Test Case 1**

**Template:**

Input:

**PlatformID** = 5

**AppVersion** = 1

**PlatformType** = AnyThing

**Expected Result**: Hello OLD

Reason:

Test case for arbitrarily high value of PlatformID > 4. The expected result must be some message as given in the problem.

**Test Case 2**

**Template:**

Input:

**PlatformID** = 0

**AppVersion** = 1

**PlatformType** = AnyThing

**Expected Result**: Hello OLD

Reason:

Test case for arbitrarily high value of PlatformID < 1. The expected result must be some message as given in the problem.

**Test Case 3**

**Template:**

Input:

**PlatformID** = 2/3

**AppVersion** = Anyvalue

**PlatformType** = AnyThing

**Expected Result**: Hello OLD

Reason:

Test case for special case handling of PlatformID = 2 or 3.

**Test Case 2**

**Template:**

Input:

**PlatformID** = 1/4

**AppVersion** = 2

**PlatformType** = AnyThing

**Expected Result**: Hello OLD

Reason:

Test case to cover the message output for AppVersion’s value below VERSION\_THRESHOLD keeping PlatformID to 1 or 4.

**Test Case 3**

**Template:**

Input:

**PlatformID** = 1/4

**AppVersion** = 3

**PlatformType** = Anything

**Expected Result**: Hello NEW

Reason:

Test case to cover the message output for AppVersion’s value above or equal to VERSION\_THRESHOLD keeping PlatformID to 1 or 4.

**Test Case 6**

**Template:**

Input:

**PlatformID** = 1

**AppVersion** = 1

**PlatformType** = SpecialDevice

**Expected Result**: Hello Old

Reason:

Test Case to cover to special handling for old version compatibility with PlatformType as SpecialDevice.