JAVA INTRODUCTION TO TESTING

Objective

The objectives of this practical session are to get you started on unit testing with JUnit.

Overview

In this lab, we are going to use JUnit to create unit tests.

You'll test a method that checks if userID and password parameters are valid by following a set of rules.

Working with Security class

1. Open the **Labs** project and add a new package called **exercise11**.
2. Add a new class to this package called **Security** (with no main())
3. Add the following code to the Security class:

```
public class Security {

 public boolean logon(String userId, String password) {
    String errorMessage;

    // userId must not be null or spaces. Must be 8 chars minimum.
    // password must contain a digit and an upper case char

    if (userId == null || userId.trim().equals("")) {
        errorMessage = "UserID may not be null or empty";
        return false;
    }
    else if (password == null || password.trim().equals("")) {
        errorMessage = "Password may not be null or empty";
        return false;
    }
    else if (!validatePassword(password)) {
        errorMessage = "Password must be min 8 chars with an
        uppercase and a number";
        return false;
    }
    else {
        errorMessage = "";
        return true;
    }
}

private boolean validatePassword(String password) {
    boolean hasUpper = false, hasDigit = false;

    if (password.length() < 8)
        return false;

    for (char c : password.toCharArray()) {

        if (Character.isUpperCase(c)) {
           hasUpper = true;
```

```
                // continue loop as no character can be both!
                continue;
            }
            if (Character.isDigit(c)) {
                hasDigit = true;
            }
        }
        return hasUpper && hasDigit;
    }
}
```

You will now test the above logon method against a set of rules which are mentioned as comments in code.

4. First you will create a test case class.
   Right-click on the package name and select the following options:
   New **>** Other **>** JUnit **>** JUnit test case.

   You need an example of what a test method looks like before producing your own test methods.

   Replace the test case in this class with the following:

```
@Test
public void testLogonEmptyUserId() {
    String userId = "";
    String password = "Freddy99";
    Security s = new Security();
    boolean actual = s.logon(userId, password);
    boolean expected = false;
    assertEquals(expected, actual);
    // assertFalse(actual);   // can also use this assert
}
```

5. Select **Run** to run this test.

6. Copy and paste the above method and create additional tests.

7. What tests would you devise? See the comments for other criteria.
   The method names must describe what the test is about.

   You will not be testing the error messages.

8. Select **Run** to run your tests.