

# BSC Software Development One

## Worksheet One

This worksheet is to help you become familiar with the basics of

- writing unit tests
- writing code to satisfy the tests

This worksheet is worth 6% of your CA

|                                     |
|-------------------------------------|
| Submit before <b>23/02/24 23:59</b> |
|-------------------------------------|

**Create a new project in Eclipse(or other IDE) called WorksheetOne\_<studentnumber>**

- With <studentnumber> being your student number i.e. **WorksheetOne\_323212**.
- I use a script to unzip folders so please be accurate with the naming.

**Create a new package with the name griffith.**

### **Task One**

#### **Part One**

Create a JUnit test file called “CalculatorTest.java” and a java file “Calculator.java”. In your JUnit test add the following four stub methods

- testAdd()
- testSubtract()
- testMultiply()
- testDivide()

In the regular java file add in the following stub methods

A stub method is a method without implementation, just the header and any default return statements.

- int add(int a, int b)
- int subtract(int a, int b)
- int multiply(int a, int b)
- double divide(int a, int b)

**(10 marks)**

#### **Part Two**

- Write test cases for each of the unit test methods that call the appropriate methods in the java file.
- The unit tests should still fail at this point as you are writing the unit test first
- Make sure to test a range of values, at least three different sets of values per test.

**(15 marks)**

### Part Three

- Implement the add method in Calculator.java
- Once its implemented test if it passes the unit test
- If it passes implement the subtract method.
- Continue, on until all methods are implemented you may only move onto the next implementation once each unit test has been satisfied.

**(10 marks)**

## Task Two

### Part One

Create a JUnit test file called "WordTest.java" and a java file "Word.java".

In your JUnit test add the following stub methods

- testContains()
- testLength()
- testNotNull()

In the Word java file add an attribute char[] attribute letters.

Add a constructor Word(char[] letters) and the following stub methods

- boolean contains(char symbol)
- int length()
- char[] getLetters()

**(10 marks)**

### Part Two

- Write test cases for each of the unit test methods that call the appropriate methods in the java file.
- The unit tests should still fail at this point as you are writing the unit test first
- Make sure to test a range of values, at least three different sets of values per test. For testContains use the assertTrue method, for testNotNull use the assertNotNull method.

**(25 marks)**

### Part Three

- Implement the methods in Word
- Remember move onto the next implementation only once each unit test has been satisfied.
- Add code in your constructor to make sure your code will always pass testNotNull.

**(20 marks)**

## **Comments and Coding Standards**

### **Comments**

- Include a comment at the top of the file with your name and student number.
- Add a comment for each coding task explaining what the code does.

**(5 marks)**

### **Follow coding standards:**

- Lowercase for variable names.
- Uppercase for the class name.
- All uppercase for constants.
- Use naming standards.
- Indent your code.

**(5 marks)**

**Zip the whole project folder and upload your folder on Moodle as WorksheetOne\_StudentNumber.zip.**