

# SD2 - Worksheet 1 - 6%



Student name:	Enkhbaatar Idersaikhan					
Student number:	3112121					
Faculty:	Computing Science					
Course:	BSCH/BSCO/EXCH			Stage/year:	2	
Subject:	Software Development 2					
Study Mode:	Full time	<input checked="" type="checkbox"/>		Part-time	<input type="checkbox"/>	
Lecturer Name:	Gemma Deery					
Assignment Title:	Worksheet 1					
Date due:						
Date submitted:						

## Plagiarism disclaimer:

*I understand that plagiarism is a serious offence and have read and understood the college policy on plagiarism. I also understand that I may receive a mark of zero if I have not identified and properly attributed sources which have been used, referred to, or have in any way influenced the preparation of this assignment, or if I have knowingly allowed others to plagiarise my work in this way.*

*I hereby certify that this assignment is my own work, based on my personal study and/or research, and that I have acknowledged all material and sources used in its preparation. I also certify that the assignment has not previously been submitted for assessment and that I have not copied in part or whole or otherwise plagiarised the work of anyone else, including other students.*

Signed: Usfj

Date: \_\_\_\_\_

Please note: [Students MUST retain a hard / soft copy of ALL assignments as well as a receipt issued and signed by a member of Faculty as proof of submission.](#)

Please do not delete the questions.

For each question insert your answer below the question

Enkhbaatar Idersaikhan 3112121

## Tasks: 1

### Part One

```
GradesTest.java  Grades.java  Calculator.java  WordTest.java  Word.java
1 //Enkhbaatar Idersaikhan 3112121
2 package griffith;
3
4 public class Grades{
5
6     public int gradesMax(int[]grades) {
7         return 0;
8     }
9
10    public int gradesTotal(int[]grades) {
11        return 0;
12    }
13
14    public double gradesAverage(int[]grades) {
15        return 0.0;
16    }
17
18    public int countFails(int[]grades, int minGrade) {
19        return 0;
20    }
21
22
23 }
```

## SD2 - Worksheet 1 - 6%

```
GradesTest.java X Grades.java Calculator.java WordTest.java Word.java
1 //Enkhbaatar Idersaikhan 3112121
2 package griffith;
3
4 import static org.junit.jupiter.api.Assertions.*;
5
6
7
8 class GradesTest {
9
10     Calculator calculator = new Calculator();
11
12     @Test
13     public void testGradesMax() {
14         //testing with positive values
15         int[] grades1 = {70, 85, 92};
16         assertEquals(92, calculator.gradesMax(grades1));
17
18         //testing with negative values
19         int[] grades2 = {-10, -5, -20};
20         assertEquals(-5, calculator.gradesMax(grades2));
21
22         //testing with null values
23         int[] grades3 = {};
24         assertEquals(0, calculator.gradesMax(grades3));
25     }
26
27     @Test
28     public void testGradesTotal() {
29         //testing with positive values
30         int[] grades1 = {70, 85, 90};
31         assertEquals(245, calculator.gradesTotal(grades1));
32
33         //testing with negative values
34         int[] grades2 = {-10, -5, -20};
35         assertEquals(-35, calculator.gradesTotal(grades2));
36
37         //testing with null values
38         int[] grades3 = {};
39         assertEquals(0, calculator.gradesTotal(grades3));
40     }
41 }
```

## SD2 - Worksheet 1 - 6%

```
41
42● @Test
43 public void testGradesAverage() {
44     //testing with positive values
45     int[] grades1 = {70, 80, 90};
46     assertEquals(80.0, calculator.gradesAverage(grades1));
47
48     //testing with negative values
49     int[] grades2 = {-10, -30, -20};
50     assertEquals(-20.0, calculator.gradesAverage(grades2));
51
52     //testing with null
53     int[] grades3 = {};
54     assertEquals(0.0, calculator.gradesAverage(grades3));
55
56 }
57
58● @Test
59 public void testCountFails() {
60     //testing pass grades and fail grades together
61     int[] grades1 = {55, 65, 75, 45};
62     int minGrade1 = 59;
63     assertEquals(2, calculator.countFails(grades1, minGrade1));
64
65     //testing only pass grades
66     int[] grades2 = {70, 80, 100};
67     int minGrade2 = 59;
68     assertEquals(0, calculator.countFails(grades2, minGrade2));
69
70     //testing only fail grades
71     int [] grades3 = {40, 45, 58};
72     int minGrade3 = 59;
73     assertEquals(3, calculator.countFails(grades3, minGrade3));
74
75 }
76 }
77
```

## Part 3

```

GradesTest.java  Grades.java  Calculator.java X  WordTest.java  Word.java
1 //Enkhbaatar Idersaikhan 3112121
2 package griffith;
3
4 public class Calculator {
5
6     // returns max grade
7     public int gradesMax(int[] grades) {
8         if(grades == null) {
9             throw new IllegalArgumentException("Grades array cannot be null or empty.");
10        }
11        //returning 0 for empty array
12        if(grades.length == 0) {
13            return 0;
14        }
15        int max = grades[0];
16        for(int grade : grades) {
17            if(grade>max) {
18                max = grade;
19            }
20        }
21        return max;
22    }
23
24    // returns total of the grades
25    public int gradesTotal(int[]grades) {
26        if(grades == null) {
27            throw new IllegalArgumentException("Grades array cannot be null");
28        }
29        int total = 0;
30        for(int grade : grades) {
31            total +=grade;
32        }
33        return total;
34    }
35
36    //returns the average of the grades
37    public double gradesAverage(int[]grades) {
38        if(grades == null) {
39            throw new IllegalArgumentException("Grades array cannot be null");
40        }
41        if(grades.length == 0) {
42            return 0.0;
43        }
44        return (double)gradesTotal(grades) / grades.length;
45    }
46
47    //returns the count of fails
48    public int countFails(int [] grades, int minGrade) {
49        if(grades == null) {
50            throw new IllegalArgumentException("Grades array cannot be null");
51        }
52        int count = 0;
53        for(int grade : grades) {
54            if(grade<minGrade) {
55                count++;
56            }
57        }
58        return count;
59    }
60 }

```

# SD2 - Worksheet 1 - 6%

## TEST

The screenshot shows an IDE with the JUnit runner interface. The top bar indicates the test is finished after 0.092 seconds. Below this, a summary shows 4/4 runs, 0 errors, and 0 failures. A green progress bar is visible. The test results list shows four successful tests: testCountFails() (0.013 s), testGradesMax() (0.001 s), testGradesTotal() (0.001 s), and testGradesAverage() (0.001 s). The right pane shows the source code for GradesTest.java, which includes package declarations, imports, and class definitions.

Project Explorer JUnit X

Finished after 0.092 seconds

Runs: 4/4 Errors: 0 Failures: 0

GradesTest [Runner: JUnit 5] (0.017 s)

- testCountFails() (0.013 s)
- testGradesMax() (0.001 s)
- testGradesTotal() (0.001 s)
- testGradesAverage() (0.001 s)

GradesTest.java X

```
1 package ...
2
3 import s...
6
7 class Gr...
8
9 Calc...
10
11 @Tes...
12 publ...
13
```

## Task 2.

```
GradesTest.java  Grades.java  Calculator.java  WordTest.java  Word.java X
1 //Enkhbaatar Idersaikhan 3112121
2 package griffith;
3
4 public class Word {
5
6     //attribute
7     private char[] letters;
8
9     //constructor that makes sure letters is never null
10 public Word(char[] letters) {
11     if(letters == null) {
12         this.letters = new char[0];
13     }else {
14         this.letters = letters;
15     }
16 }
17 //Returns true if the letters array contains the gi
18 public boolean contains(char symbol) {
19     for(char letter : letters) {
20         if(letter == symbol) {
21             return true;
22         }
23     }
24     return false;
25 }
26 //Returns the number of letters
27 public int length() {
28     return letters.length;
29 }
30 //returns the letters array
31 public char[] getLetters() {
32     return letters;
33 }
34
35 }
```

## SD2 - Worksheet 1 - 6%

```
GradesTest.java  Grades.java  Calculator.java  WordTest.java  Word.java
1 //Enkhbaatar Idersaikhan 3112121
2 package griffith;
3
4 import static org.junit.jupiter.api.Assertions.*;
5
6
7
8 class WordTest {
9
10     @Test
11     public void testContains() {
12
13         //word contains the char
14         char[] letters1 = {'h','e','l','l','o'};
15         Word word1 = new Word(letters1);
16         assertTrue(word1.contains('e'));
17
18         //word doesn't contains the char
19         char[] letters2 = {'w','o','r','l','d'};
20         Word word2 = new Word(letters2);
21         assertFalse(word2.contains('e'));
22
23         //char repeated multiple times
24         char[] letters3 = {'m','o','o','n'};
25         Word word3 = new Word(letters3);
26         assertTrue(word3.contains('o'));
27     }
28
29     @Test
30     public void testLength() {
31
32         //testing normal word
33         char[] letters1 = {'m','o','n','g','o','l','i','a'};
34         Word word1 = new Word(letters1);
35         assertEquals(8,word1.length());
36
37         //testing 1 letter
38         char[] letters2 = {'f'};
39         Word word2 = new Word(letters2);
40         assertEquals(1,word2.length());
41
42         //testing empty array
43         char[] letters3 = {};
44         Word word3 = new Word(letters3);
45         assertEquals(0,word3.length());
46     }
47
48     @Test
49     public void testNotNull() {
```



## SD2 - Worksheet 1 - 6%

```
48 @Test
49 public void testNotNull() {
50
51     //normal letters array
52     char[] letters1 = {'t','e','s','t'};
53     Word word1 = new Word(letters1);
54     assertNotNull(word1.getLetters());
55
56     //passing a null should still result in a non-null letters array
57     Word word2 = new Word(null);
58     assertNotNull(word2.getLetters());
59
60     //empty array
61     char[] letters3 = {};
62     Word word3 = new Word(letters3);
63     assertNotNull(word3.getLetters());
64 }
65 }
66
```

## TEST

The screenshot shows an IDE with the JUnit test runner. The left pane displays the Project Explorer and the JUnit test results. The right pane shows the source code of WordTest.java.

**JUnit Test Results:**

- Finished after 0.088 seconds
- Runs: 3/3
- Errors: 0
- Failures: 0
- WordTest [Runner: JUnit 5] (0.017 s)
  - testContains() (0.013 s)
  - testLength() (0.001 s)
  - testNotNull() (0.001 s)

**WordTest.java Source Code:**

```
20 Word word2 = new Word(letters2);
21 assertFalse(word2.contains('e'));
22
23 //char repeated multiple times
24 char[] letters3 = {'m','o','o','n'};
25 Word word3 = new Word(letters3);
26 assertTrue(word3.contains('o'));
27
28
29 @Test
30 public void testLength() {
31
32     //testing normal word
33     char[] letters1 = {'m','o','n','g','o'}
```

My user handle for GitHub is : SylerEdd

GitHub link to worksheet 1: <https://github.com/SylerEdd/JUnit>