Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

Assignment Cover Sheet

Learner Name:

Saul Agustin Güemes Enriquez

Learner Number:

3175794

Faculty:

BS Computing Science

Programme: Computing Science Stage/Year: 2

Module:

Software Development 2

Study Mode: Full Time **X** Part-time _____

Lecturer Name:

Haseeb Younis

Assignment Title:

BSC_SD2_WorksheetTwo_3175794_SaulAgustin_GüemesEnriquez

Additional Relevant Information (e.g. number of pieces submitted etc.):

I need to get my invitation accepted. Please email saulagustin.gaemesenriquez@student.griffith.ie when the invitation is accepted.

AI in Learner Assessment Policy Indicate here applicable categories allowed for this assignment.	Category of AI Use	Allowed or not allowed
	No AI Use	
	AI for Planning	
	AI for Editing	
	AI for Support Tasks	
	AI for Collaboration	
	Full AI Use	**(NOT ALLOWED)**

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

Academic Integrity Honour Code:
I submit this work in line with the principles of Academic Integrity as appearing in the Academic Integrity policy, the assessment description(s), and the relevant AI Assessment Scale in the AI in Learner Assessment

I affirm that I have not given or received any *unauthorised* help, from a person or through unauthorised content generation on this assignment, and that this work is my own.

I understand that penalties may be imposed if this assessment is in breach of the Academic Integrity and Misconduct policy.

Signature: Saul Date: February

Please note: Learners **MUST** retain a hard/soft copy of all assignments and must have the lecturer/member of Faculty acknowledge/sign as proof of submission.

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

Task

- Create two files “ConversionsTest.java” and “Conversions.java” and commit them to the repository.

- In Conversions.java add method stubs and only the stubs for the methods below, once you have written all stubs commit them to the repository. Do not implement the methods until after the tests are written.
-
    - public double euroToDollar(double euro)
    - public double dollarToEuro(double dollar)
    - public int stringToInteger (String val)
    - public String integerToString (int val)
    - public String switchCase() // change uppercase to lowercase and vice versa
    - Include screenshots of the method stubs below

```
public class Conversions {  no usages  new *
    public double euroToDollar(double euro) {  n
        return 0;
    }

    public double dollarToEuro(double dollar) {
        return 0;
    }

    public int stringToInteger(String val) {  no
        return 0;
    }

    public String integerToString(int val) {  no
        return null;
    }

    public String switchCase() {  no usages  new *
        return null;
    }
}
```

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

- In ConversionsTest.java add Unit tests for each method.
  - Each unit test must perform at least three tests on the method involved.
  - Test different ranges (positive, negative, null values, etc..)
  - Write them one at a time as you will need to commit after writing each
  - For each individual unit test, after writing the tests, run it and it should fail, then commit it to the repository.
  - In Conversions.java implement the appropriate method so it will pass the unit test, once it succeeds commit it to the repository.
  - do not move onto the next test until the previous one has succeeded and has been committed.
  - Continue until all tests have been completed.
  - For each unit test include screenshots below of
    - the unit test
    - the tests failing
    - the method implementation
    - the tests succeeding

- euroToDollar

the unit test

```
@Test
public void euroToDollar() {
    // positive
    assertEquals( expected: 11.0, converter.euroToDollar(10.0), delta: 0.01);
    // negative
    assertEquals( expected: -5.5, converter.euroToDollar(-5.0), delta: 0.01);
    // zero
    assertEquals( expected: 0.0, converter.euroToDollar(0.0), delta: 0.01);
}
```

the tests failing

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

```
/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java ...

java.lang.AssertionError:
Expected :11.0
Actual   :0.0
<Click to see difference>

> <1 internal line>
>     at org.junit.Assert.failNotEquals(Assert.java:835) <2 internal lines>
>     at ConversionsTest.euroToDollar(ConversionsTest.java:12) <25 internal lines>


Process finished with exit code 255
```

the method implementation

```java
public double euroToDollar(double euro) {
    // random conversion rate
    return euro * 1.1;
}
```

the tests succeeding

```
✓ Tests passed: 1 of 1 test – 2 ms

/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

- dollarToEuro

the unit test

```
// test dollarToEuro
@Test  new *
public void dollarToEuro() {
    // positive
    assertEquals( expected: 10.0, converter.dollarToEuro(11.0),  delta: 0.01);
    // negative
    assertEquals( expected: -5.0, converter.dollarToEuro(-5.5),  delta: 0.01);
    // zero
    assertEquals( expected: 0.0, converter.dollarToEuro(0.0),  delta: 0.01);
}
```

the tests failing

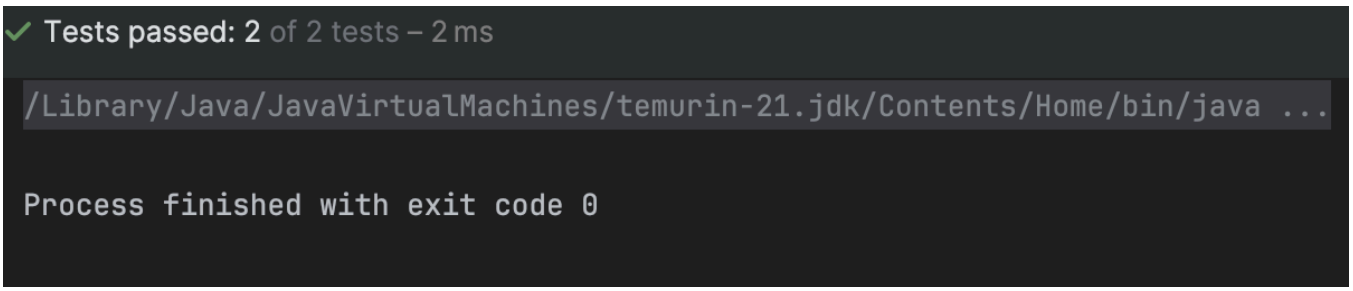
```
/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java ...

java.lang.AssertionError:
Expected :10.0
Actual   :0.0
<Click to see difference>
```

the method implementation

```
public double dollarToEuro(double dollar) {
    return dollar / 1.1;
}
```

the tests succeeding

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

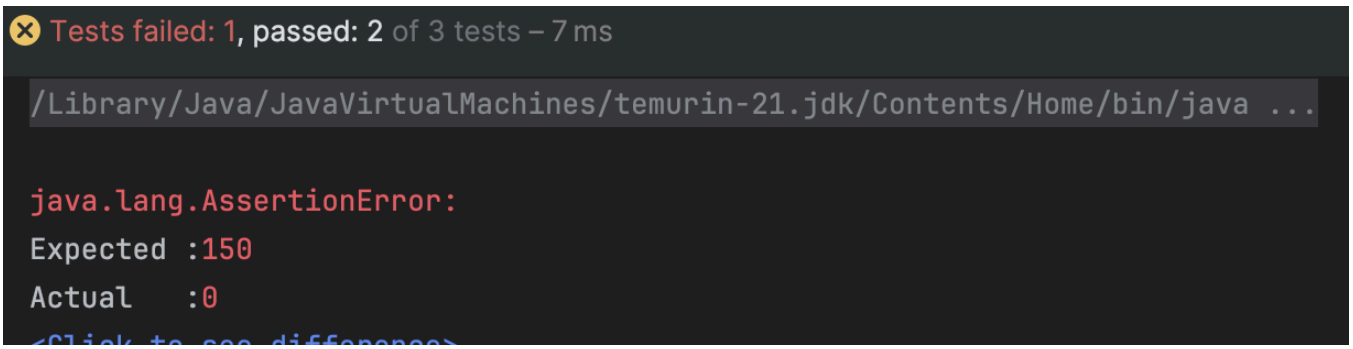


- stringToInteger

the unit test

```
// test stringToInteger - pos, neg, null
@Test  saulagus
public void stringToInteger() {
    // positive string
    assertEquals( expected: 150, converter.stringToInteger( val: "150"));
    // negative string
    assertEquals( expected: -42, converter.stringToInteger( val: "-42"));
    // null string
    assertThrows(NumberFormatException.class, () -> {
        converter.stringToInteger( val: null);
    });
}
```

the tests failing



Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

the method implementation

```java
public int stringToInteger(String val) {  3 usages
    // check for null or empty string
    if (val == null || val.isEmpty()) {
        throw new NumberFormatException();
    } else {
        // convert string to integer
        return Integer.parseInt(val);
    }
}
```

the tests succeeding

```
✓ Tests passed: 3 of 3 tests – 3 ms

/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

- integerToString

the unit test

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

```java
// test integerToString - pos, neg, zero
@Test  saulagus
public void integerToString() {
    // positive
    assertEquals( expected: "99", converter.integerToString( val: 99));
    // negative
    assertEquals( expected: "-99", converter.integerToString( val: -99));
    // zero
    assertEquals( expected: "0", converter.integerToString( val: 0));
}
```

the tests failing

```
Tests failed: 1, passed: 3 of 4 tests – 7 ms
/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java ...

org.junit.ComparisonFailure:
Expected :99
Actual   :
```

the method implementation

```java
public String integerToString(int val) {
    // convert integer to string
    return String.valueOf(val);
}
```

the tests succeeding

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

```java
public String integerToString(int val)
    // convert integer to string
    return String.valueOf(val);
}
```

- switchCase

  the unit test

```java
    // test switchCase - mixed, empty, null
    @Test  new *
    public void switchCase() {
        // mixed
        assertEquals( expected: "hELLO", converter.switchCase( val: "Hello"));
        // empty
        assertEquals( expected: "", converter.switchCase( val: ""));
        // null
        assertNull(converter.switchCase( val: null));
    }

}
```

  the tests failing

```
Tests failed: 1 of 1 test – 7 ms
/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java ...

org.junit.ComparisonFailure:
Expected :hELLO
Actual   :
<Click to see difference>
```

  the method implementation

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

```java
public String switchCase(String val) {  3 usages  saulagus *
    // check for null
    if (val == null ) return null;
    // check for empty string
    if (val.isEmpty()) return "";
    // convert string to char array
    char[] valArr = val.toCharArray();
    // loop through char array
    for (int i = 0; i < valArr.length; i++) {
        // check if char is uppercase
        if (Character.isUpperCase(valArr[i]))
            valArr[i] = Character.toLowerCase(valArr[i]);
        else
            valArr[i] = Character.toUpperCase(valArr[i]);
    }
    // convert char array to string
    return new String(valArr);
}
```

the tests succeeding

```
✓ Tests passed: 1 of 1 test – 2 ms

/Library/Java/JavaVirtualMachines/temurin-21.jdk/Contents/Home/bin/java ...

Process finished with exit code 0
```

Marks

- Unit tests **(25 marks)**
- Implementation **(25 marks)**
- Comments and Coding Standards **(10 marks)**
- Repository with proper commit history **(40 marks)**

Comments and Coding Standards

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

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.

Follow coding standards:

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

Zip the whole project folder and upload your folder on Moodle as SD2_LabTwo_StudentNumber.zip.

Note that it is an absolute requirement of the worksheet that you commit after each completion of a unit test and implementation. You will lose marks if your repository does not reflect this.

Learning Outcomes

This assignment is assessing the following programme and related module learning outcomes: Programme and related module learning outcomes that this assignment is assessing.

1. Install, configure, and utilize a testing framework for a software project. 2. Use technical design and implementation skills.

Artificial Intelligence Assessment Scale (AI AS):

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

AI Assessment Scale Framework:

https://www.griffith.ie/sites/default/files/2025-05/f-6.5-ai

in-learner-assessment-policy.pdf"

This assignment is subject to AI permission option:
AI for Editing [X]
Assignment Specific AI Instructions:

- AI may be used to assist in bug fixing or to check for issues in your code before submission.
- AI may not be used to generate project code or comments. All submitted code should be your own work.
- If unauthorized or undeclared AI use is found, you will be required to explain your submission to show understanding of the code.

Please note:

- All AI generated content and process must be declared.
- Any task that is not listed as permitted may be assumed to be prohibited. - If it is determined that a learner has used an AI tool in any way that is not permitted by the chosen level, then they will be subject to the disciplinary procedures listed in the Academic Integrity and Misconduct policy regarding the misuse of AI in assignments.

Penalties

Late Penalties:
Standard penalties will be applied to work that is submitted late, as per faculty guidelines. https://moodle.griffith.ie/mod/resource/view.php?id=18202
Penalties that apply to this assignment:
All code should be submitted as .java files.
Submission of text files, .class files or other file types will incur a 20% penalty.
Submission of file types that cannot be graded such as .jar files will result in resubmission, late penalties will apply to any resubmissions.
Not providing access to git repository will result in a loss of marks.
Uncommented code may result in no marks.
AI generated comments and code may result in no marks.

Assessment Criteria

Saul Agustin Güemes Enriquez - 3175794
https://github.com/saulagus/SD2_LabTwo_3175794
Repo shared to haseeb.younis@griffith.ie. As of Feb 19, pending invite acceptance.

Assessment criteria applied to this assignment:

Conversions Class stubs **10**
Test at least 3 sets of values for each ConversionTests unit tests **25**
Implementation for each method **15**

Repository
Git repo shared **5**
Commit with stubs **5**
Commit for each unit test 10
Commit for each implementation 10
Commit in correct order stub->unit test->implementation-> next unit test **10**

Comments 5
Coding standards 5

Total 100

Submission Instructions and Plagiarism

Plagiarism:
All work must be your own. Any cases of suspected plagiarism will be thoroughly investigated and may be brought in for VIVA. All parties involved in cases of plagiarism will receive a 0 and their student record marked accordingly. Any submission that is found to be plagiarised will receive a grade of 0 without the possibility of resubmission. ***By submitting your assignment you accept that you understand what plagiarism is and that your assignment is not plagiarised in any way.***

Instructions for submission:
All work must be submitted via the submission point on Moodle, email submissions not accepted.
Submit all files as a zip file to Moodle in the format
SD2_LabTwo_number.zip
e.g. SD2_LabTwo_1000323