

**National College of Ireland**  
**BSc. (Hons) in Computing, Year 2, BSHCIFSC2**

**Software Quality and Testing**

**Semester 2, 2023-24**

**CA3 [18%]**

---

**Notes:**

- **This is an open book assessment.** Any resources used in answering the questions should be referenced in the submitted document, and listed in the *References* section.
- The question (and all sub-questions of that question) that you have to attempt is assigned based on **the last digit of your student ID number**. Please carefully read the *Question To Be Attempted* guidelines to identify the question assigned to you. **This is a submission requirement**. If the incorrect question is answered, no marks will be provided for that question.
- **This is an individual assessment.** All work submitted **should be your own** and should be carried out using only the **concepts covered in this module**. Conferring with others **is not permitted**.
- The responses should include all the intermediate steps/computations for obtaining the answer.
- **YOU ARE NOT ALLOWED TO PUBLISH THIS ASSIGNMENT BRIEF OR A PART THEREOF ON ANY WEBSITES. YOU ARE NOT ALLOWED TO PUBLISH/SHARE YOUR SOLUTION WITH OTHERS. Note that you ARE NOT ALLOWED TO USE ANY generative AI tools (e.g., ChatGPT, GitHub Copilot, etc.)**
- Note that all submissions will be electronically screened (via Turnitin) for evidence of academic misconduct (e.g., plagiarism, collusion, AI generated text and/or code, etc.).
- You should submit your answers electronically in a .docx or .pdf document via the *CA3 Submission Link (Report part/tab)* available on the module's Moodle page. You are also required to submit all the .java files corresponding to the implementation of your unit tests and functionalities tested as a .zip file via the *CA3 Submission Link (Source Code part/tab)* available on the module's Moodle page. **These are submission requirements.**
- **Attachments (available on Moodle)**  
Java source code for the calc.java and calc.zip are available on the module's Moodle page in the files:
  - calc.java

**Assignment Submission Deadline Date/Time: 17<sup>th</sup> April 2024**

## ***Question To Be Attempted***

The question (and all sub-questions of that question) that is to be attempted for the assessment is assigned based on **the last digit of your student ID number** as described below.

**This is a submission requirement.** If the incorrect question is answered, **no marks** will be provided for that question.

Question	Last Digit of Student ID	Question to Attempt
Question 1	0, 2, 4, 6, 8	1 A.
	1, 3, 5, 7, 9	1 B.

Example: A student with the student ID = 2365432**1** would answer the *Question 1 B.* (because the last digit of that student ID is **1**).

**Software Quality and Testing**  
**Continuous Assessment 3 [18%]**  
**Unit Testing**

As part of the continuous assessment for the Software Quality and Testing module, you are required to answer the questions assigned to you and submit the answers electronically in a .docx or .pdf document via CA3 Submission Link (Report part/tab) available on the module's Moodle page. You are also required to submit all the .java files corresponding to the implementation of your unit tests and functionalities tested as a .zip file via the CA3 Submission Link (Source Code part/tab) available on the module's Moodle page.

You are required to implement and execute JUnit unit tests in NetBeans IDE and answers the questions assigned to you in a document.

**Question 1**

1 A . Let us consider a class that implements the basic calculator operation. *Figure 1* presents a possible implementation. Note that the source code is available on the module's Moodle page in the file *Calc.java*. The add method is provided, you can modify the class by adding methods to allow the multiply, subtraction and division operations too.

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 * this license
 */
```

```
package com.mycompany.calc;
```

```
/**
 *
 * @author tj
 */
public class Calc {
```

```
    public static int add(int a, int b){
```

```
        int c=a+b;
        System.out.println(c);
        return c;
    }
```

```
    public static void main(String[] args) {
        int b=3, c=5;
        int a=add(b, c);
        System.out.println("Hello World!" + a);

    }
}
```

*Figure 1 Source code for Question 1 A.*

- a) Implement all the required unit tests in order to check whether the compute method works according to expectation, namely according to the specification provided in *Question 1 A*
- b) Run/execute the unit tests implemented at item a) and provide in your report a screenshot of the NetBeans *Test Results Window* that includes a summary of the execution of the unit tests (namely, for each unit test, the name of the test method run and whether the test has passed or failed).
- c) Implement a method for computing area of circle. Then implement the required unit test
  - i. Update the test
  - ii. Include a screenshot of the source code corresponding to latest version of your methods in the report.

**[100 marks]**

**Continues on next page**

1 B . Let us consider a class that implements the basic calculator operation. *Figure 2* presents a possible implementation. Note that the source code is available on the module's Moodle page in the file *Calc.java*. The add method is provided, you can modify the class by adding methods to allow the multiply, subtraction and division operations too.

```

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change
 * this license
 */

package com.mycompany.calc;

/**
 *
 * @author tj
 */
public class Calc {

    public static int add(int a, int b){

        int c=a+b;
        System.out.println(c);
        return c;
    }

    public static void main(String[] args) {
        int b=3, c=5;
        int a=add(b, c);
        System.out.println("Hello World!" + a);

    }
}

```

*Figure 2 Source code for Question 1 B.*

- a) Implement all the required unit tests in order to check whether the compute method works according to expectation, namely according to the specification provided in *Question 1 B*.
- b) Run/execute the unit tests implemented at item *a)* and provide in your report a screenshot of the NetBeans *Test Results Window* that includes a summary of the execution of the unit tests (namely, for each unit test, the name of the test method run and whether the test has passed or failed).
- c) Implement a method for computing circumference of circle. Then implement the required unit test
  - i. Update the test
  - ii. Include a screenshot of the source code corresponding to latest version of your methods in the report.

**[100 marks]**

