

**National College of Ireland**  
**BSc. (Hons) in Computing, Year 2, BSHCIFSC2\_B**  
**BSc. (Hons) in Computing, Year 2, BSHCIFSC2\_ColaisteDhulaigh**  
**BSc. (Hons) in Computing, Year 2, BSHIFSC2\_Rathmines**  
**Higher Certificate in Computing, Year 2, HCCOMP2**

**Software Quality and Testing**

**Semester 2, 2023-24**

**CA2 [16%]**

---

**Notes:**

- **Open book assessment.** Any resources used in answering the questions should be referenced in the submitted document, and listed in the *References* section.
- **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**.
- **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, Google Copilot, etc.).**
- Note that all submissions will be electronically screened (via Turnitin) for evidence of academic misconduct (i.e., plagiarism and collusion).
- You should submit your answers in a .docx or .pdf document electronically via the *CA2 Submission Link* available on the module's Moodle page

**Assignment Submission Deadline Date/Time: 12<sup>th</sup> March 2024 at 16:45**

## ***Questions To Be Attempted***

The questions that are to be attempted for assessment are based on the last digit of your student ID number as described below.

This is a submission requirement. If the incorrect questions are answered, no marks will be provided for those particular questions.

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

**Software Quality and Testing**  
**Continuous Assessment 2 [16%]**  
**Statement Coverage. Branch Coverage.**

As part of the continuous assessment for the Software Quality and Testing module, you are required to answer the questions below as part of the lab session, and submit the answers electronically in a single .docx or .pdf document via the *CA2 Submission Link* available on the module's Moodle page.

**Question 1**

1. A. Let us consider the method in *Figure 1. A.*:

```
3      public boolean isSumMultipleOfSix(int value, int anotherValue) {  
4          int sum = value + anotherValue;  
5          int remainder = sum % 6;  
6          boolean isMultipleOfSix;  
7          if (remainder == 0) {  
8              isMultipleOfSix = true;  
9          } else {  
10             isMultipleOfSix = false;  
11         }  
12         return isMultipleOfSix;  
13     }
```

*Figure 1.A Question 1.A. method*

- a) Using the statement coverage technique, identify suitable test cases (i.e., test input data) to achieve full (i.e., complete) code coverage.
- b) What is the code coverage calculated based on the statement coverage technique for the test cases you have identified at item a)? The responses should include all the intermediate calculations for obtaining the answer.

**[40 marks]**

## Question 1

1. B. Let us consider the method in *Figure 1. B*:

```
3      public boolean isSumMultipleOfNine(int value, int anotherValue) {
4          int sum = value + anotherValue;
5          int remainder = sum % 9;
6          boolean isMultipleOfNine;
7          if (remainder != 0) {
8              isMultipleOfNine = false;
9          } else {
10             isMultipleOfNine = true;
11         }
12         return isMultipleOfNine;
13     }
```

*Figure 2.B Question 1.B method*

- Using the statement coverage technique, identify suitable test cases (i.e., test input data) to achieve full (i.e., complete) code coverage.
- What is the code coverage calculated based on the statement coverage technique for the test cases you have identified at item a)? The responses should include all the intermediate calculations for obtaining the answer.

**[40 marks]**

## Question 2

2. A. Let us consider the method in Figure 2.A.:

```
3      public void calculate(int a, int b, int c){
4          int result = 0;
5          if (a >= 3 || c < -1){
6              result = 2 * a * c;
7              System.out.println("result: " + result);
8          }
9          System.out.println("result: " + result);
10         if (a > 9 && b > 2){
11             result = a - b + c;
12             System.out.println("result: " + result);
13         }
14     }
```

Figure 2.A Question 2.A method

- Create/draw the control flow graph for the given method (alternatively, you can create the flow chart for the given method)
- Using the branch coverage technique identify suitable test cases to achieve full code coverage.
- What is the code coverage calculated based on the branch coverage technique for the test cases you have identified at item b)? The responses should include all the intermediate calculations for obtaining the answer.

[60 marks]

## Question 2

2. B. Let us consider the method in Figure 2.B.:

```
3      public void calculate(int x, int y, int z) {
4          int r = 0;
5          if (x > 2 && y == 6){
6              r = z/x;
7              System.out.println("r is: " + r);
8          }
9          System.out.println("r is: " + r);
10         if (x == 7 || z < 8){
11             r = z - y + x;
12             System.out.println("r is: " + r);
13         }
14     }
```

Figure 2.B Question 2.B method

- Create/draw the control flow graph for the given method (alternatively, you can create the flow chart for the given method)
- Using the branch coverage technique identify suitable test cases to achieve full code coverage.
- What is the code coverage calculated based on the branch coverage technique for the test cases you have identified at item b)? The responses should include all the intermediate calculations for obtaining the answer.

[60 marks]