

University of Lincoln Assessment Framework

Assessment Briefing Template 2021-2022

Module Code & Title:

CMP1903M Object Oriented Programming

Contribution to Final Module Mark:

30%

Description of Assessment Task and Purpose:

This is Assessment 1 and is an **individual** assignment.

This assignment looks at part of the process in implementing a problem – **a code review**. A code review is a review of your code by another developer or developers. Code reviews can help with:

Motivation

Sharing best practice

Also, they can highlight:

Accidental/structural errors

Legibility

Even short, informal code reviews can have a great impact on code quality and error frequency.

You should for this assignment use **one** of these methods:

1.

- a) Grab the 'base' code from Blackboard
- b) Create a Github repo and host the code
- c) Create a 'Development' branch
- d) Modify the base code and add functionality
- e) Submit the code (which is described below) to your colleagues for peer review through a **pull request**
- f) Merge the modified code back into your 'Main' branch.
- g) Provide *helpful* reviews on the other students code
- h) Review at least 2, receive at least 2 reviews
- i) Complete the self-assessment checklist

2. a) Grab the 'base' code from Blackboard

b) Create a Github repo and host the code

c) Modify the base code and add functionality

d) Reviewers will raise an 'issue'

e) Exchange code review comments through the issue

f) Review at least 2, receive at least 2 reviews

g) Complete the self-assessment checklist.

3. Exchange emails to review the code. You can host your code on Github, or just email it for review. Add your modifications to the base code.

You should review at least 2 and receive at least 2 reviews of your code.

Complete the self-assessment checklist.

The questions which you should ask in your review are:

- 1) Is the code properly documented/commented?
 - a. Does it need more work?/Is it great?
 - b. Where should it be better?
- 2) Does the code handle errors properly?
 - a. Does it need more work?/Is it great?
 - b. Where could it be better?
- 3) What suggestions do you have to improve the code?

Please see the **Criterion Reference Grid** for details of how the presentation will be graded.

Learning Outcomes Assessed:

- [LO1] Demonstrate the use of version control tools in a software development project
- [LO3] Apply object-oriented principles to the implementation of software programs

Knowledge & Skills Assessed:

Subject Specific Knowledge, Skills and Understanding:

Code review process;

C# programming;

Professional Graduate Skills:

Code review and assessment skills

Emotional Intelligence:

Awareness of others, managing and supporting others, motivation, self-confidence.

Career-focused Skills:

Code review.

Assessment Submission Instructions:

Submit your code to Blackboard in Assessment 1 Supporting Documents.

You should also submit to Assignment 1 Upload:

- a. Submit a brief description of the code review process [LO1]:
 - i. Who did you provide reviews for.
 - ii. How many reviews you received, and who were they from
 - iii. Were the reviews you received helpful/not helpful?
 - iv. What did you consider changing/modifying after the reviews?
 - v. Show evidence of the reviews (screenshots are ok)
- b. This brief description should be around 600 words
- c. Include after the description above, the self-assessment checklist [LO3].

Your code review document should be submitted through Blackboard in the '**CMP1903M Assessment Item 1 Upload**' section of the Assessments folder before the deadline for the assessment.

Date for Return of Feedback:

Please see the School assessment dates spreadsheet.

Format for Assessment:

Written code review document submitted to Blackboard.

Feedback Format:

Written Blackboard feedback.

Additional Information for Completion of Assessment:

The coding task: Analysis of Text

Analysis of text is a crucial aspect of many computing applications. The ability to single out words and phrases allows us to look for, for example: themes, sentiment and complexity of language. For example, many on-line games allow players to message each other during the game and twitter users send different messages. Analysis of these messages might reveal a lot about the writers and their experience of the application they are using.

In this part of the assignment you are required to create a programme that looks at sentences of text, use appropriate data structures to store the words and carry out an analysis of the words entered.

On running the programme the user should be presented with two options in the console:

1. Do you want to enter the text via the keyboard?
2. Do you want to read in the text from a file?

If '**Option 1**' is selected then the user will enter **one or more** sentences of text, one sentence at a time. The use of an asterisk (*) could indicate the end of the entry. For example they may enter:

```
>The cat sat on the mat.*
```

The program should then report back some basic analysis of this text:

Number of sentences entered = 1

Number of vowels = 6

Number of consonants = 11

Number of upper case letters = 1

Number of lower case letters = 16

The frequency of individual letters = ?

If **'Option 2'** is selected then a file of pre-written text is opened, read and a similar analysis is carried out and displayed on the console. In addition a file of "long words" is created as part of the application and saved in the current directory i.e. any words longer than 7 characters.

A pre-set example text file is provided, along with the expected values for the questions above to allow you to test your code to some degree.

In your code:

- There should be at least the methods indicated.
- There should be at least the custom classes indicated.
- There should be examples of exception handling.

Assessment Support Information:

Performing code reviews: <https://bit.ly/3qVeCwR>

Important Information on Dishonesty & Plagiarism:

University of Lincoln Regulations define plagiarism as 'the passing off of another person's thoughts, ideas, writings or images as one's own...Examples of plagiarism include the unacknowledged use of another person's material whether in original or summary form. Plagiarism also includes the copying of another student's work'.

Plagiarism is a serious offence and is treated by the University as a form of academic dishonesty. Students are directed to the University Regulations for details of the procedures and penalties involved.

For further information, see www.plagiarism.org