Programme: BSc Digital and Technology Solutions **Level:** 2

Module and Module Code: UFCFYM-15-2 Object-oriented Software Design & Development II

UWE Credit Rating: 15 **Module Leader:** Jack Kimmins

Assessment: Assessment B

Assessment Type: Practical Portfolio **Word Count/Duration:** No word count/duration

Submission Date: See Assessment Plan **Handout Date:** See Assessment Plan

Submission Method: Teams Submission **Feedback Due Within:** 20 working days

Assessment Title: OOSDDII A2 – Practical Portfolio

This assessment covers 50% of the assessment for this module.

MODULE OUTCOMES COVERED IN THIS ASSESSMENT:

Module Outcomes

- **MO1**: Apply procedural and/or object-oriented programming techniques.
- **MO4:** Implement, test, and debug complex software solutions to meet a requirements specification.
- **MO5:** Develop complex software solutions and software modifications to specified requirements.
- **MO6:** Test code and analyse results to correct errors found using unit testing.
- **MO7:** Apply underlying concepts and the principles of best practices and standards.

Format: See the programme and module guides for further guidance as to the standard format for this type of assessment.

Note: Students' attention is drawn to the penalties attached to any form of assessment offence. In completing the assessment front sheet, you are declaring that the work is entirely your own. References used in the assessment must be referenced according to the UWE Harvard System, as outlined in the Library Services' iSkillZone.

Grade	Descriptions
90-100%	Meeting all the requirements for the 89.9% mark and in addition demonstrating a creative and unique synthesis of ideas and concepts including an evaluation of the methodological approach adopted
83-89.9%	Comprehensive coverage of all criteria for assessment, all of which have been explicitly interpreted. Extensive interrelating of alternative viewpoints or resources with the development of novel or original ideas and in-depth reflection. Access to, and judicious selection from, a wide range of sources, many of which are original. Analysis and evaluation clearly and appropriately expressed demonstrating a confident integration of appropriate ideas and concepts in a succinct and elegant manner. Opportunities taken to refer to and engage critically with module learning outcomes and their implications as appropriate to the submission. The work demonstrates the student's ability to engage with appropriate dimensions of genre and discourse
76-82.9%	Comprehensive coverage of all criteria for assessment, many of which have been explicitly interpreted. Extensive interrelating of alternative viewpoints or resources with the introduction of novel or original ideas and in-depth reflection. Access to, and judicious selection from, a wide range of sources, many of which are original. Analysis and evaluation clearly and appropriately expressed, and a confident discussion of ideas and texts is demonstrated. Opportunities taken to refer to and engage critically with module learning outcomes.
70-75.9%	Comprehensive coverage of all criteria for assessment, many of which have been explicitly interpreted. Extensive interrelating of alternative viewpoints or resources with abundant evidence of reflection. Access to, and judicious selection from, a wide range of sources, many of which are original. Analysis and evaluation clearly and appropriately expressed. Opportunities taken to refer to and engage critically with module learning outcomes.
67-69.9%	Extensive coverage of all criteria for assessment with sound interpretation apparent. Main issues or principles are clearly elaborated in clear, cogent, and reflective argument. A good range of sources utilised, and the use of primary sources is prioritised. Good analysis and evaluation coherently and fluently expressed demonstrating a scholarly presentation of ideas and an astute sense of audience. Opportunities taken to refer to module learning outcomes.
63-66.9%	Extensive coverage of all criteria for assessment with some interpretation apparent. Main issues or principles clearly elaborated and with creditable degree of reflection. A good range of sources utilised, with some being original. Good analysis and evaluation coherently and fluently expressed. Opportunities taken to refer to module learning outcomes.
60-62.9%	Coverage of all criteria for assessment with some interpretation apparent. Main issues or principles clearly elaborated and with creditable degree of reflection. A good range of sources utilised, with some being original. Good analysis and evaluation coherently and fluently expressed. Opportunities taken to refer to module learning outcomes.
57-59.9%	Largely relevant coverage of the criteria for assessment with the main issues or principles identified explicitly. A satisfactory level of reflectivity upon a range of sources, all relevant but mostly secondary. Analysis and/or evaluation attempted with success. Module learning outcomes acknowledged with some implications reviewed.
54-56.9%	Largely relevant coverage of the criteria for assessment with the main issues or principles identified explicitly. A satisfactory level of reflectivity upon a range of sources, all relevant but mostly secondary. Analysis and/or evaluation attempted with moderate success. Module learning outcomes acknowledged with some implications reviewed.
50-53.9%	Largely relevant coverage of the main criteria for assessment and a satisfactory level of reflectivity upon a range of sources that are largely relevant but mostly secondary. Some attempt at analysis with moderate success. Module learning outcomes acknowledged, and some implications reviewed.
47-49.9%	Adequate relevant coverage of the criteria for assessment with some development of the criteria but little interpretation apparent. Some irrelevance and/or inaccuracies in the selection of content. Reflection apparent with beginning of development but erratic analysis and evaluation. Level of study insular.
44-46.9%	Adequate relevant coverage of the main criteria for assessment. Some irrelevance and/or inaccuracies in the selection of content. Reflection apparent but minimal analysis and evaluation. Module learning outcomes referred to appropriately. Level of study insular.
40-43.9%	Barely satisfactory coverage of the criteria for assessment with a substantial degree of irrelevance and inaccuracy apparent. Reflectivity acknowledged but lacking substance. Little analysis and evaluation. Use of secondary sources only and little more than acknowledgment of the module learning outcomes.
Fail	
35-39.9%	Unsatisfactory coverage of the criteria for assessment with a substantial degree of irrelevance and inaccuracy apparent. Reflectivity acknowledged but lacking in substance. A marked absence of analysis and evaluation. Use of secondary sources only and little acknowledgement of the module learning outcomes.
28-34.9%	Failure to meet most of the stated criteria with the work largely irrelevant to the assignment set. Inappropriate reference to the literature and/or relevant experience.
1-27.9%	Failure to meet most of the stated criteria with the work largely irrelevant to the assignment set. Inappropriate reference to the literature and/or relevant experience.
0%	Failure through non-submission.

Marking Criteria and Distribution

The following table outlines the weighting of specific tasks/module outcomes. Please ensure focus effectively correlates following percentage weighting.

Module Outcome	Criteria	Related Task	Percentage Weighting
1, 4	Write an efficient and effective object-oriented program in C# .NET 6 that implements the solution, you must also ensure that your code is fully documented.	1	30%
7	Conforming to industry standards.	1	20%
1, 4, 5	Providing additional relevant functionality to the system.	1	10%
6, 4	Test the effectiveness of your application by creating a test plan and supporting evidence of testing, for example, screenshots etc.	1	10%
1, 5, 7	Evaluation & Code Review	2	30%

Please note that marking/grades will consider the following:

- Academic writing skills.
- Range of referenced materials used.
- Correctness and accuracy of UWE Harvard referencing.
- Presentation of content and structure.
- Presentation skills (Presentations only).

Assessment Introduction

Your task is to create a C# object-oriented solution that not only satisfies the given requirements but also showcases your understanding of industry best practices and standards in design, programming, testing, and evaluation.

Key Assessment Notes

Assignment Submission Guidelines:

For your final submission, please organise your project into three distinct and clearly labelled folders. Exclude compiled binaries, cache, or temporary files. Once organised, compress all relevant files (evaluation documents, source code, database exports etc.) into a single ZIP file. Name the ZIP file using the following convention: WSxxxxxx_OOP2.zip

Industry Standards:

An integral part of Task 2 revolves around how well you adhere to established industry standards in your project. These standards are not merely 'nice-to-haves' but are essential attributes that make your code more readable, maintainable, and professional.

Industry standards could include:

- Code Commenting (Inline Comments, Docstrings/Comment Blocks)
- Abstraction
- Appropriate Variable Names and Standardised Naming Conventions
- Indentation and White Space
- Encapsulation
- Appropriate Code Structures
- Inheritance and Polymorphic Behaviours
- Modularisation, Testability and Reusability
- Implementation Security (Sanitise inputs, secure password storage etc.)
- Code Efficiency
- Error Handling
- Code Documentation

Your ability to effectively apply these industry standards in your programming will significantly impact the marks awarded for this assessment.

Additional Recommended Reading/Resources:

Please refer to the module handbook for the full recommended reading list.

- UWE Online Library - http://www1.uwe.ac.uk/library/
- jCode Library - https://jcode.stablenetwork.uk/
- w3Schools - https://www.w3schools.com/
- LinkedIn Learning - https://www.linkedin.com/learning/

Task 1: Morse Code Translator (60%)

Task Introduction

After your recent success with the previous tasks in OOSDDI, you have been contracted as a software engineer by the UK's Ministry of Defence. They have requested you develop a fast and secure console-based Morse Code Translator that will facilitate rapid Morse code translations and interpretations between central command and the troops on the ground. The MOD already has the infrastructure in place for radio communication, your program will instead be responsible for what is transmitted and received over this communication network.

Your translator application will need to be able to do all the following:

- Interpret text into and from several different types of Morse code standards.
- Log translations and conversions for archival purposes (for example, using a local file system or a relational database etc.)
- Utilise secure cryptography methods to ensure the privacy of all data being processed by your solution.

Program Requirements

Morse Code Standards

Currently, the MOD want the program to support the following Morse code standards but would like to easily be able to expand this list if required:

- International Morse Code
- American Morse Code

Note: In addition to this brief, your client has provided you with three text files containing character encodings for each of the standards. These must be programmatically imported into the application and to remain compliant with their internal policies, the contents of the files must not be altered (including removing comments, empty lines etc.)

Application Flow

Upon execution, the user should be presented with a menu for navigation of the various functions of your application. The user should be able to select a standard, input text and for it to be converted into Morse code. The input should utilise input validation, for example, entered characters that did not exist in the selected Morse code character set should result in an action (error message, switch to valid characters etc.).

Upon a successful translation, the translation should be logged and displayed to the user in this example format:

•••• •– ••• – •–|•–•• •–|•••– •• ••• – •–|–••• •– –••• –•––

Spaces between letters and | between words.

There should also be a reverse option, there will enable users to enter Morse code and a standard for it to be interpreted back into readable text.

In addition, the user should be able to use encrypted versions of the Morse code standards that utilise secure cryptographic cyphers (AES, Blowfish, Triple DES etc.). The process should be very similar to direct text conversion, but the user will be required to enter a password that will be used as an encryption/decryption key.

Finally, the MOD would like your program to have a training feature. This break-out section from the main menu will enable the user to select a standard and then be randomly shown a morse-code encoded character from that set. The user is then required to guess what character this corresponds with. The user should be informed if they were correct/incorrect, and the process should repeat with a new character. This will keep happening until the user requests to go back to the main menu.

Testing Requirements

Test the effectiveness of your application by creating a test plan and supporting evidence of testing i.e., screenshots. Your test plan should contain the following attributes:

- Test Description
- Test Data
- Expected Outcome
- Actual Outcome
- Evidence

Additionally, the use of unit tests could be used to further validate the successful implementation of your code. These tests should be included in your project's solution and uploaded in your submission.

Requirements Checklist

- ✓ **Additional: Program Authentication**
- ✓ **Importing Morse Standards from Text Files**
- ✓ **Main Menu**
  - **Direct Morse Code Encoding/Decoding**
    - **Select Standard**
    - **Enter Text**
    - **Input Validation**
    - **Output and Logging**
  - **Direct Encrypted Morse Code Encoding/Decoding**
    - **Select Standard**
    - **Enter Text**
    - **Input Validation**
    - **Enter Encryption Key**
    - **Output and Logging**
  - **Additional: Option for Compression/Decompression**
  - **User Training**
  - **User Accounts Management**
    - **Create New Users**
    - **Edit Existing Users**
    - **Deleting Users**
    - **Changes should be saved and loaded upon program restart.**
- ✓ **Modularity**
  - **File Splitting**
  - **Use of functions/procedures/methods**
- ✓ **Additional: Creative and Unique Synthesis of Additional Program Functionality**
- ✓ **Application Testing**
  - **Test Plan**
    - **Both White and Black Box Testing**
    - **Failed Tests**
      - **Re-attempting Tests**
  - **Unit Testing**
- ✓ **Code Commenting**

Task 2: Evaluation & Code Review (30%)

Once you have completed your portfolio, you will be asked to present your work and demonstrate your understanding of the project and content to your module lead. This demonstration, lasting up to 20 minutes, can be conducted in person or, under exceptional circumstances, online.

The demonstration will be divided into two parts:

1) The first part will focus on your portfolio. You will be asked to explain your solution's source code and answer questions about your implementation.
2) The second part will involve pre-written questions about the content covered throughout the module.

The purpose of this demonstration is not just to showcase your final product, but also to assess your understanding of the principles and practices you've learned during the module. Be prepared to answer questions related to your project and the overall module content.

[END OF ASSESSMENT BRIEF]