

<b>Title of Module</b>	<b>U10787 Application Development in C#</b>
<b>Title(s) of Assignment</b>	<b>Interactive math platform and Fishing Quota</b>
<b>Module Team</b>	<b>Dr. Alexios Louridas Mr. Gordon Aiken (Moderator)</b>
<b>Assignment Deadline</b>	<b>16 May 2023 at 14:00</b>
<b>1st Progress Check Deadline</b>	<b>24 March 2023</b>
<b>2nd Progress Check Deadline</b>	<b>28 April 2023</b>
<b>Assignment Weighting</b>	<b>100%</b>
<b>Expected Feedback Date</b>	<b>Throughout the semester with final feedback on the 6 June 2023</b>
<b>Location of Feedback</b>	<b>Via GitHub and Turnitin on Blackboard</b>
<b>Assessment Type</b>	<b>Portfolio (Solution in C# and a written report)</b>
<b>Where to Submit</b>	<b>GitHub and Turnitin [1]</b>
<b>What to Submit</b>	<b>A Visual Studio 2022 solution under version control on GitHub, and a report submitted in Turnitin. Alternative a zip format compressed folder of a complete project directory with a 'Git' local repository and a report submitted in Turnitin.</b>

[1] If you experience any problems with the turnitin system then please contact the Computing Administration Team (computing@canterbury.ac.uk)  
else if you experience any problems with the Git or GitHub system then please contact any member of the module team

## 1 Overview

The aim of this assignment is to produce a software solutions to solve the outlined problems below. This project would need to be completed over this semester with assistance week by week while covering important topics using assessment for learning. It is therefore very important you attend each week to learn and to perform.

The Classroom link to join the assignment is:

<https://classroom.github.com/a/nhZ9Fx59>

You are a member of a team of software developers of the infamous company Bennett Sarres Software Solutions Ltd., and working to design two distinct applications.

- An interactive math solver.
- Fishing fleet quota.

### 1.1 Application 1 - Interactive Math Solver

An interactive math solver is a tool or software that can solve mathematical problems, often including equations and inequalities, by interacting with the user. It may provide step-by-step explanations, graph visualizations, and other features that can help users understand and solve the problem. Some interactive math solvers are designed for specific purposes, such as solving calculus problems or solving systems of equations, while others are more general-purpose.

Some interactive math solvers are available online, while others can be downloaded or installed on a computer. You are going to create the latter type.

## **1.2 Application 2 - Fishing fleet quota**

The fishing fleet quota in the United Kingdom is a system that regulates the amount of fish that can be caught by UK fishing vessels. The quota is set by the European Union (EU) and is designed to prevent overfishing and ensure the sustainable management of fish stocks. Under the quota system, each fishing vessel is allocated a certain amount of fish that it is allowed to catch, based on the total allowable catch (TAC) for each species. The quota is allocated to individual vessels or to fishing fleets, and can be bought, sold, or leased among vessels.

As of 2021, the UK has left the European Union, and has taken back control over its waters. The UK government has announced that it will establish new quota management system, which will allow the UK to set its own quotas and decide its own fishing limits. These regulations have now been updated with the latest quota management rules being the 2021 November update.

It is important to note that the fishing quota system is a complex and constantly changing topic, with negotiations between different countries and potential future changes.

## **1.3 Expected Outcome**

You are required to use the industry-standard tool Git in combination with GitHub for your repository service and also for issue tracking and project planning. You shall need to be active throughout the whole course in order to be able to keep up with the project end result. There are going to be progress checks throughout the development process of the applications where you would be able to obtain verbal feedback.

To be more specific you shall be marked according to the following points:

- Project Progress - Planning and Issue Tracking
- User Interface
- Coding and Design

For more information about the marking see the Mark Rubric Scheme on page [13](#).

## 2 Assignment Instructions

The portfolio would follow the following progress checks:

### 2.1 Phase 1 [10%] 24th March 2023

In this phase you would be checked on the commits and progress of the interactive math solver application.

### 2.2 Phase 2 [10%] 28th April 2023

In this phase you would be checked on the commits and progress of the Fishing fleet quota application.

### SRS documentation for Fishing Fleet Quota application[10%]

Create a Software Requirements Specification (SRS) document for **only** the Fishing Fleet Quota application before you start coding. In the document you should follow the template provided in BlackBoard. Most of the sections can be filled using bullet points. Your documentation submission should be done in **TurnItIn via BlackBoard**.

### 2.3 Applications

#### 2.3.1 Interactive Math Solver [30%]

You are required to create a windows form application with appropriate controls. Make sure you select appropriate controls depending on what you are trying to achieve. Do not use controls in ways that the user does not expect for them to be used. Make sure that your application has an easy flow and a user does not require a manual for its operation.

When writing code make sure that once it works, you try and optimise it as much as possible. Also, make sure you follow coding standards throughout your code.

The application shall require to:

- The interactive math solver shall be a windows form application
- The application user interface shall have a clean and intuitive interface that is easy for users to navigate and understand.
- The application shall have the following problem-solving capabilities:
  - Find GCD of multiple numbers
  - Find LCM of multiple numbers
  - Solve Linear equations with double coefficients
  - Solve Quadratic equations with double coefficients
  - Find the determinant of any integer matrix up to 4x4.
  - Solve a system of linear equations of 2 or 3 unknowns using Cramer's rule. The linear equations should only have integer coefficients

### 2.3.2 Fishing Fleet Quota [40%]

You are required to create a windows form application with appropriate controls. Make sure you select appropriate controls depending on what you are trying to achieve. Do not use controls in ways that the user does not expect for them to be used. Make sure that your application has an easy flow and a user does not require a manual for its operation.

When writing code make sure that once it works, you try and optimise it as much as possible. Also, make sure you follow coding standards throughout your code.

**Customer requirement:** “Before I go to my boat fleet and head out to go and fish, I always check with my local contacts and find out what species of fish they are in need and how many. Then I check the quota limit for these species of fish. Having this, I head to my fleet and board my boat. I might be days at sea, though, and some of my fleet boats will send me daily information on how much they caught on that day. I have a PC running windows 10 at my boat, I would like to have a program that I can pass the catch of the day for each fish species, add how many my fleet caught every day and making sure that I have not exceeded my quota. Also, I need to know for each boat what they actually have at the end of each day. This way I can quickly communicate to my boats of my fleet to stop fishing certain species or finish and go back to shore to unload.”

**Software manager requirements:** The application shall require to:

- Add new boats in the fleet

When adding new boats, they should have the following specified:

- Name
- License number (License numbering format allowed: L XXX, LL XXX, L XX, LL XX)
  - \* Where L is a letter and where X is a number.
- Maximum load size in kg or tonnes
- Fish species they can store (Number of species of fish allowed, minimum 2 and maximum 4)

- Have the ability for a user to select a specific boat.

When adding new boats, they should have the following specified:

- Add fish caught in kg for each boat in the fleet
- See the name of the boat, the species of fish they are allowed to fish, the live weight they are currently carrying for each species of fish.
- Display a quota report to show the total weight caught by the fleet for each fish species versus the quota. (Example: Cod caught = 2.3tonnes Cod catch limit = 4 tonnes)
- Have the ability for a user to select an option to display any of the above weights in kg or tonnes.

## 2.4 Version Control

Once you have joined and accepted the assignment in GitHub you will need to clone the following repository.

<https://classroom.github.com/a/nhZ9Fx59>

In the repository you will find a solution with two projects one for each application.

It would be good practice to create appropriate issues in GitHub that could be linked to your requirements. Try and make reasonable branches for multiple similar issues and try and merge to master main when appropriate. Your final master main branch would count as your final submission thus ensure that your latest branch is merged to your master main. There shall also be an additional submission point in blackboard if required. **Note that marks are awarded for the usage of Git and GitHub. If your final submission is only in blackboard no marks would be awarded towards the version control mark scheme.**

For more information about the marking see the Mark Rubric Scheme on page [13](#).

### 3 What are catch limits?

Catch limits are measured in tonnes (t) or kilograms (kg). They are the maximum live weight that may be caught and retained on board. You calculate the live weight by multiplying weight by the suitable conversion factor.

Table 1: Conversion Factors

Species	Factor
Angler	1.22
Cod	1.17
Haddock	1.17
Hake	1.11
Horse mackerel	1.08
Witches	1.06
Plaice	1.05
Skates and rays	1.13
Whiting	1.18

Information was taken from: [Commercial fishing and fisheries government page](#)

Table 2: Current quota limits for each species

Stock (Species)	Catch limit (tonnes)
Angler	8
Cod	1
Haddock	3.5
Hake	15
Horse mackerel	0.5
Witches	1.5
Plaice	8
Skates and rays	2
Whiting	3

Information was taken from: [Current catch limits 10 metres and under pool 2023 government page](#)

## **4 Submission Guidelines**

Your submission should happen no later than the **16<sup>th</sup> of May 2023 at 14:00**.

The final code that would be marked will be the one in the final push to the *master* branch, or the code submitted in the turnitin bucket whichever is the latest.

Your documentation should also be submitted in Turnitin. Please make sure you submit a pdf or any type of word document. Any other formats that cannot be read will be disregarded.

Please note that a single large (most of your project) or multiple commits on the last day (few hours before deadline) would be detrimental in your marking. The reason being that performing all coding in a few hours would indicate that you have done everything in a rush and thus undermining ethical issues while coding, which means that learning outcomes would not be completely satisfied.

### **4.1 CCCU Late Submission Guidelines**

In the case of late submission of work for a component of assessment (including Git commits, wiki documentation, dissertations, extended essays and projects) without an approved extenuating circumstance, the work will be penalized.

The standard penalty will apply to all components of assessed work for all programmes, unless:

**EITHER:**

- (i) a special regulation has been approved by Academic Board;

**OR:**

- (ii) the work is marked on a pass/fail basis and it is not possible to give a numeric mark.

The standard penalty to be applied by all programmes will be 5% of the eligible marks, per day, for up to 7 days, after which a mark of 0 will be recorded.

The reduction of 5% will be applied to the total mark that the student can receive for the component of assessment and not to the mark that the student has earned. If, therefore, the component of assessment is marked out of 100 and the student has scored 50%, the student will lose 5 marks and receive 45%.

## **5 Formatting of Submission**

You are expected to produce a single solution in visual studio 2022. It is expected that the solution should contain the projects containing the applications. There is no target length of your documentation but it should not be beyond 2400 words. Any screen shots, tables, figures, charts, illustrations, etc. will not contribute towards the word count. Your work must be adequately referenced throughout, using Harvard referencing style. Pears & Shields (2016) give a complete guide to Harvard referencing. Guidelines on using the Harvard Referencing style are available at:

<https://www.canterbury.ac.uk/library/docs/harvard.pdf>

<https://www.canterbury.ac.uk/students/docs/study-skills/resource-1-Harvard-Referencing-Guide.pdf>

The report must be submitted using the dedicated Blackboard grade centre submission bucket on or before the submission deadline.



## 6 Resources You May Find Useful

For your help there are Unit tests written to make sure the interactive math solver methods you write run correctly. This will help you verify your code. Although extremely care was taken for these unit tests to run as expected beware that there is a possibility of error. If you believe there is something please let the module team know.

[To see how to run the unit tests click here.](#)

[Example of Math Calculators](#)

[Cramer's Rule Explained](#)

[Understand fisheries catch limits and closed fishing areas](#)

[IEEE Software Requirements Specification \(SRS\) Template](#)

[SRS Example of Template](#)

## **7 Learning Outcomes Assessed (Fully or Partially)**

<b>LO</b>	<b>Description</b>
1.	Use a data and process driven mechanism to design and create simple solutions to problem situations using Graphical User Interface based solutions;
2.	Have an understanding and utilise effectively simple guideline approaches to good GUI design;
3.	Use simple programming constructs effectively, show a development in using more complex constructs and use objects from existing classes.

## **8 Graduate/Employability Skills Gained**

- Adaptable
- Digitally Literate
  - Confident in use of digital devices, applications and services. In particular, developing new services using existing C# application and the use of professional tools required in the workplace..
- Effective Communicator
  - Using interactions with customers and product manager to develop the best product possible.
- Informed
  - Aware of issues relating to social justice, ethics and wellbeing. In particular, ethics and law of using IP material belonging to other parties. Viewing of a sample the License Agreement used in industry. Use of technology to help prevent harm to others. help prevent harm to others.
- Innovative
- Professional
- Self-Aware

## **9 Programmes of Study**

BSc (Hons) Computer Science  
BEng (Hons) Software Engineering  
BSc (Hons) Computing  
BSc (Hons) Computer Forensics and Security

## APPENDIX A: MARK RUBRIC

Table 3: Rubric for Phase 1 Progress Check.

Marks	Typified by — Rubric for Phase 1 Progress Check
0 - 3	No progress or very poor progress. No or few commits, no or little project planning, no or little coding.
4 - 6	Some progress in the code with phase 1 started and in good level. Few well commented commits and good project planning.
7 - 10	Excellent progress in the code with phase 1 nearly completed. Excellent to good use of tools for project planning.

Table 4: Rubric for Phase 2 Progress Check.

Marks	Typified by — Rubric for Phase 2 Progress Check
0 - 3	No progress or very poor progress. No or few commits, no or little project planning, no or little coding.
4 - 6	Some progress in the code with phase 1 started and in good level. Few well commented commits and good project planning.
7 - 10	Excellent progress in the code with phase 1 nearly completed. Excellent to good use of tools for project planning.

Table 5: Rubric for documentation.

Marks	Typified by — Rubric for documentation
0 - 3	No document submitted or one that does not present enough information about the application.
4 - 5	Basic documentation with few information about the application requirements and how it works.
6 - 7	Documentation that includes all sections with relevant information but not gone detailed enough.
8 - 10	An excellent document with all sections having detailed information.

Table 6: Rubric for Interactive Math Solver Application.

Marks	Typified by — Rubric for Interactive Math Solver Application
0 - 11	Program does not compile or poor user interface that is impossible to use without training. Few commits that are not well thought and bad comments. Coding is scarce and badly implemented.
12 - 14	Basic GUI, but lacks finesse, lack of error handling, poor alignment of information / components etc. Implementation issues may be present. Basic algorithms to perform tasks, but overly complex, contain errors or inefficient. Basic coding standards apparent. May have some inconsistency. Some commits but with not appropriate comments.
15 - 20	Good UI that fits purpose and shows nice design elements. Responsibilities appropriately allocated and distributed. Good use of algorithms to solve tasks. Minor errors in algorithms, with some performance issues. Good coding standard. No inconsistency. Good number of commits with some good comments.
21 - 30	Excellent GUI with good usage of layouts and components. Good templates and showing some data binding. A good program design for the application, code wise. Excellent coding standard. No inconsistency. Very good commits and with good comments.

Table 7: Rubric for Fishing Fleet Quota Application.

Marks	Typified by — Rubric for Fishing Fleet Quota Application
0 - 15	Program does not compile or poor user interface that is impossible to use without training. Few commits that are not well thought and bad comments. Coding is scarce and badly implemented.
16 - 19	Basic GUI, but lacks finesse, lack of error handling, poor alignment of information/ components etc. No classes used, basic responsibilities given to form methods. Implementation issues may be present. Basic algorithms to perform tasks, but overly complex, contain errors or inefficient. Basic coding standards apparent. May have some inconsistency. Some commits but with not appropriate comments.
20 - 23	Good UI that fits purpose and shows nice design elements. Use of Classes with Classes not fully completed. Responsibilities appropriately allocated and distributed within classes. Good use of algorithms to solve tasks. Minor errors in algorithms, with some performance issues. Good coding standard. No inconsistency. Good number of commits with some good comments.
24 - 27	Excellent GUI with good usage of layouts and components. Very good use of classes to solve the issue with appropriate responsibilities implemented. Few, if any problems with an implementation. Good coding standard. No inconsistency. Good number of commits with some good comments.
28 - 40	Excellent GUI with good usage of layouts and components. Very good use of classes to solve the issue with appropriate responsibilities implemented separating back end from GUI. Few, if any problems with an implementation. Excellent coding standard. No inconsistency. Very good commits and with good comments.