Department of Computer Science	**UCLan Coursework Assessment Brief**	2023-2024
	Module Title: Programming	
	Module Code: CO1409	Level 4
	Pizza Ordering Application	This assessment is worth 40% of the overall module mark

THE BRIEF / INSTRUCTIONS

For this assignment, you are tasked with creating a pizza ordering application **using C++**.

This is an individual project, and no group work is permitted.

You will be assessed on your implementation of the solution, which **must be produced using C++**, as well as your understanding of your code.

Do not diverge from the assignment specification. If you do not conform to the assignment specification then you will **lose marks**.

You may conduct your own research into topics that have not been explicitly taught within the module (this will be required for higher marks) however, you should include a justification for its use alongside links to any sources in your program comments. **Failure to do so may result in a plagiarism investigation. Only use concepts that you are confident you understand.**

You must upload your assignment code, along with the code from your Summary Exercise in Workbook 2. Failure to do so will result in a reduction of your mark.

You must also attend your demonstration during your lab (see below). Failure to do so will result in a mark of 0.

Learning Outcomes
This assessment has been designed to assess the following learning outcomes:

- Apply the principles of programming.
- Design an appropriate solution for a given problem.
- Implement a readable and maintainable software solution based on their own design.
- Evaluate the quality of his or her developed software.

.

Implementation

You have been tasked with developing a pizza ordering application for the Student Centre café. You are required to implement a proof-of-concept console-based pizza ordering application to demonstrate to the café managers.

The pizza ordering application requires the **size** of the pizza to be selected as well as **at least one topping**.
A list of available sizes and toppings is available in the table below.

The **total price** of the pizza is the **size price + the price of all toppings added**.

Pizza Size	Cost (credits)
Small	5.0
Medium	8.50
Large	10.25

Toppings	Cost (credits)
Ham	0.80
Mushrooms	0.50
Pepperoni	1.0
Peppers	0.80
Onions	0.40
Extra Cheese	1.50

Continued on the next page..

Features

As the developer, you have total control regarding the menu, interactions, flow of the application and messages displayed the user. Use the examples included in ***Appendix A: Example Outputs*** for inspiration. However, you must attempt to implement the logic for following features defined in the table that follows.

Feature	Description
Add Credits	The application will require a user to input a number of credits into the pizza ordering application. The credits will represent the currency.
Order Pizza	The user should be able to select the size of the pizza and **at least one topping.** The application should calculate the total cost of the pizza based on the size and the topping(s) the user has selected.
Checkout	If there are sufficient credits available, the user should be able to then purchase the pizza selected deducting the amount from the available credit and return the remaining credits back to the user.
Recent Orders	The user should be able view all recent orders made using the application, including the pizza size, toppings and price. **High first criteria only*

Expectations

You are expected to use and implement the programming concepts you have learned throughout this module to complete the implementation aspect of this coursework. You will also be assessed on the readability and appropriate use of programming concepts to implement features of the application defined above. This involves:

- Indenting your code blocks correctly.
- Choosing suitable data types.
- Choosing suitable names for variables
- Choose a suitable name for programming project.
- Appropriate use of programming concepts.
- Appropriate use of comments within your code that describe key or complicated aspects.

Demonstration

To accompany the application, you will be demonstrating how your application works to your lab tutor (not the whole class). During this demonstration you can expect to be asked the following areas:

- Your choice of data types
- How you have made efforts to improve the readability and maintainability of your code
- How key parts of your application work
- How you handle errors made by users

This is not an exhaustive list. Your lab tutor can ask you about anything in your program, so it is important to only use concepts you fully understand. If in doubt ask your lab tutor or the module leader.
Do not copy and paste code from online.

During your demonstration you will also be asked to show your completed Summary Exercise from Workbook 2 and answer any changes your tutor may have. Whilst this is not a test, it is important that you fully understand the work you have completed so far in order to prepare you for the following semester.

Marking Scheme

To obtain a pass mark of 42 you must:

- You must have completed Summary Exercises 1 and 2 and included the code in your submission.
- Allow users to add credits at the start of your program
- Display the list of available pizza sizes and allow users to make a selection
- Display the list of available toppings and allow the user to add **at least one** topping
- Allow users to checkout to complete their purchase which includes:
    - Calculating the total price of the pizza (size + topping price)
    - If the user has sufficient credits available to purchase the pizza, a message should be displayed stating the transaction has been accepted.
    - If the user does not have enough credits available a message should be displayed stating that there are insufficient credits available.
- You should be able to explain the code you have written for your assignment and in the submitted lab sheets during your demonstration

For Grade Bands 45 – 48 you must (in addition to the above)**:**

- Your code must be properly indented and laid out so that it is readable.
    - Brackets must line up (and should normally be on a line of their own).
    - Indentation must be consistent.
    - Appropriate use of white space should be made.
    - Over-long lines of code or comments should be split up.
- You should have no ‘magic numbers’ but instead make proper use of constants.
- Variable names should be meaningful and not excessively long.
- Your code should be commented appropriated.

For Grade Bands 52 – 55 you must (in addition to the above)**:**

- Functions should be used to separate your code into logical blocks. Your mark will be capped at 48 without appropriate use of functions within your assignment.
- Functions should be appropriately named and commented.
- Your program should have an easy-to-use menu system. If a user enters a selection which is not on the menu an error message should be displayed.
- Users should be prevented from entering negative numbers when adding credits.
- If a user does not have sufficient credits to purchase a pizza, your program should prompt the user to add more credits. Only once the user has sufficient credits can the purchase proceed.

For Grade Band 58 you must (in addition to the above)**:**

- Users should be able to add **more than one topping**.
- After purchasing a pizza your program should update the user’s credit balance and give them the option to purchase another pizza.
- During your demonstration, you should be able to answer questions about the readability and maintainability of your program.

For Grade Bands 62 – 68 you must (in addition to the above)**:**

- You must have completed Summary Exercise 3 and included the code in your submission. Marks cannot be awarded if functions have not been used within the solution for this exercise.
- Data should be passed into functions using parameters. Where necessary, appropriate return types should also be used.
- All user inputs should be fully validated. Additionally, if a number is expected then an error message should be displayed if the user enters a string.
- During your demonstration, you should be able to describe and justify why you have created functions and what they are used for.

For Grade Bands 74 – 100 you must (in addition to the above)**:**

- There should be no large blocks of repeating code within your program.
- There should be no global variables however, constants are acceptable.
- Function parameters should be passed in by reference where appropriate. You should explain your reasoning as part of your demonstration.
- Code and comments should be written in a professional manner.
- Your program should include an additional 'Recent Order' feature that recalls all of the orders (including the pizza size, toppings and price) and displays them on the screen.
  Note: this will require some independent research.
- During your demonstration you should be able to provide detailed answers to all questions and also be able to reflect on what is considered good practice. I.e. How have you avoided using global variables in your code, and why is it considered bad practice to use them?
- You should have completed the Summary Exercise 4 and corrected the program to ensure that the calculations output the correct values.

PREPARATION FOR THE ASSESSMENT

Before attempting this assessment, it is highly recommended that you revisit:

- **Lecture notes** – Any notes you took during the lectures.
- **Lab worksheets** – Read over all lab worksheets.
- **Lab projects** – Ensure all projects have at least stage one implemented.

Combined, these provide all the necessary information for you to complete this assessment successfully. All resources are available on the CO1409 Blackboard area under *Module Materials*.

RELEASE DATES AND HAND IN DEADLINE

Assessment release date:	04th April 2024
Assessment submission deadline date and time:	**30th April 2024 at 1700Hrs to Blackboard**

Please note that this is the final time you can submit – not the time to submit!
Your feedback / feed forward and mark for this assessment will be provided on 31-May-2024

SUBMISSION DETAILS

Please take your time when reading this section, as this contains specific information on how you should submit your coursework.

- **You must use C++ to develop your solution.**
- Ensure your name and Student ID number (located on the back of your UCLan card) is stated at the top of your code.
- You should submit your **entire** project folder as a zip folder.
- **Do not only submit the .sln file as your code will not be included in your submission.**
- **You must demonstrate and explain your submission to your tutor in the lab sessions.**
- **You may be asked to demonstrate and explain your submission as part of an interview.**
- The use of automated code generator tools is forbidden. Any use of these tools would result in an academic misconduct investigation.
- **Late submissions**: Except where an extension of the hand-in deadline date has been approved, work that is handed in within 5 working days late will receive a maximum mark of 40%. Work handed in later than this will receive 0%.
- **Academic Malpractice**: The consequences of academic malpractice in assessments are serious. This includes plagiarism, collusion and allowing other students to access your work. This will not be tolerated. Details surrounding the coursework regulations can be found in the University's Academic Regulations.

Below are tips that you may find useful when working on this assessment:

- Do not leave this assessment to the last minute.
- If you have any questions regarding this coursework, ask the module leader or module tutors.
- Give yourself plenty of time to submit prior to the submission deadline.
- Use pen/cil and paper to work out the flow of your application.

HELP AND SUPPORT

- Support will be provided via Microsoft Teams (CO1409 channel), and email. You will also have the opportunity to ask questions during lectures / labs.
- For support with using library resources, please contact our subject librarian subjectlibrarians@uclan.ac.uk. You will find links to lots of useful resources in the My Library tab on Blackboard.
- If you have not yet made the university aware of any disability, specific learning difficulty, long-term health or mental health condition, please let us know. The Inclusive Support team will then contact you to discuss reasonable adjustments and support relating to any disability. For more information, visit the Inclusive Support site.
- To access mental health and wellbeing support, please complete our online referral form. Alternatively, you can email wellbeing@uclan.ac.uk, call 01772 893020 or visit our UCLan Wellbeing Service pages for more information.
- If you have any other query or require further support you can contact The Student Support Centre. Speak with us for advice on accessing all the University services as well as the Library services. Whatever your query, our expert staff will be able to help and support you. For more information, how to contact us and our opening hours visit Student Support Centre.
- If you have any valid mitigating circumstances that mean you cannot meet an assessment submission deadline and you wish to request an extension, you will need to apply online prior to the deadline.

Disclaimer: The information provided in this assessment brief is correct at time of publication. In the unlikely event that any changes are deemed necessary, they will be communicated clearly via e-mail and a new version of this assessment brief will be circulated.	Version: 2 Updated 01/09/2022

APPENDICIES

Appendix A – Example Outputs

The examples below represent potential interactions and console outputs between the user and vending machine application:

Main Menu Example

```
-------------------------------
UCLan PIZZA
-------------------------------
MAIN MENU
  1. Add Credits (current credits = 0.00)
  2. Order Pizza
  0. Exit

Please enter a number: 2
```

Order Pizza – Select Size

```
ORDER PIZZA– Select Size    [Current Balance = 10.00 credits]

Please choose from the following options:

  1. Small   [3.0 credits]
  2. Medium  [5.50 credits]
  3. Large   [7.25 credits]
  0. Return to Main Menu
........................................
Please enter a number : 2
You have selected a medium pizza.
........................................
```

Order Pizza – Select Toppings

```
You have selected a medium pizza.
```

```

Please choose from the following toppings:

  1. Cheese       [0.80 credits]
  2. Salad        [0.50 credits]
  3. Bacon        [1.00 credits]
  4. Ketchup      [0.30 credits]
  5. Mayo         [0.30 credits]
  6. Extra Pizza [1.50 credits]
  0. Return to Main Menu
........................................
Please enter a number : 3
You have added bacon to your pizza. Your current price is: 6.50 credits.
Would you like to add additional toppings?
Please input 'Y' for yes and 'N' for no : N
........................................
```

Purchase (sufficient credits) Example

```
........................................
Please enter a number : 3
You have added bacon to your pizza. Your current price is: 6.50 credits.
Would you like to add additional toppings?
Please input 'Y' for yes and 'N' for no : N
........................................
Available Balance:    10.00 credits
Pizza price:          6.50 credits
----------------------------------------
Your new Balance  = 3.50 credits
----------------------------------------
Would you like to order another pizza?
Please input 'Y' for yes and 'N' for no : N
Thank you, goodbye!
```

Purchase (insufficient credits) Example

```
........................................
Please enter a number : 3
You have added bacon to your pizza. Your current price is: 6.50 credits.
Would you like to add additional toppings?
Please input 'Y' for yes and 'N' for no : N
........................................
Available Balance:    0.00 credits
Pizza price:          6.50 credits
----------------------------------------
You have insufficient credits available. You require "6.50" credits.
Would you like to add more credits?
Please enter 'Y' for yes and 'N' for no : Y
.........................................
Please enter how many credits you would like to add to your balance: 7.00
----------------------------------------
Your new Balance  = 7.00 credits
Would you like to continue processing your order?
  Please enter 'Y' for yes and 'N' : Y
........................................
Available Balance:    7.00 credits
Pizza price:          6.50 credits
----------------------------------------
Your new Balance  = 0.50 credits
----------------------------------------
Thank you, goodbye!
```