

**School of Psychology and Computer Science**

**CO1404 (CO1409) – Programming 2021/2022**

* **Assessment 1**: Coursework
* **Title:** UCLan Pizza Ordering Application

**Name:** Adnan Ali Mumtaz

**Email :** aamumtaz1@uclan.ac.uk

Table of Contents

[1 Introduction 2](#_Toc57549709)

[2 Questions & Answers 3](#_Toc57549710)

[3 Documentation 6](#_Toc57549711)

[3.1 Variables 6](#_Toc57549712)

[3.2 Methods & Functions 6](#_Toc57549713)

[3.3 User Guide 7](#_Toc57549714)

[4 Testing 8](#_Toc57549715)

[4.1 Use-case 8](#_Toc57549716)

[5 Reflection 9](#_Toc57549717)

# Introduction

The purpose of the report is overall reflection of one’s learnings from the assignments. Report runs through several stages, which aims to understand more about student learning and its assignments content such as Variables, Methods and Functions etc.

Report’s intention is to give effective insight in how the assignment has provided the knowledge and skills, which required to get assignment tasks done. This would also help student in gaining further knowledge of c++ in depth. It also focuses on programs testing phases and user guide etc.

The content of the report has been divided in several topics, which are:

* Questions & Answers
* Documentation (Variables, Methods & Functions)
* Testing (User-Case)
* Reflection

Content has been specifically designed to bring out the most valuable information possible regarding the assignment and students learnings, this would also help understand how further amendments could be made to possibly make the assignment more effective and ease in teaching the required skills and knowledge.

# Questions & Answers

Please provide your own answers to the following questions (A - E). Take note of the number of potential marks awarded for each question, as this is an indicator to how much detail you should provide.

**Question A**

|  |  |
| --- | --- |
| In C++, which of the following data types should be used to store the price of an item – i.e. 4.30  Please choose one of the following options by entering **X** into the box of your selection | |
| Options | Selection |
| string |  |
| int |  |
| double | **X** |
| bool |  |

***(1 Mark)***

**Question B**

|  |  |
| --- | --- |
| Which of these would not improve the readability of code?  Please choose one of the following options by entering X into the box of your selection | |
| Options | Selection |
| Comments |  |
| Meaningful Variable Names |  |
| Indentation |  |
| The Debugger | **X** |

***(1 Mark)***

**Question C**

|  |  |
| --- | --- |
| Examine the following code.  Which of the following should replace **<operator>** in order for **result** to have a value of **2?**         int result;         int value = 50;         if (value > 0 **<operator>** value <= 20)         {                result = 1;         }         else if (value > 20 **<operator>** value <= 50)         {                result = 2;         }         else if (value > 50 **<operator>** value < 75)         {                result = 3;         }    Please choose one of the following options by entering **X** into the box of your selection | |
| Options | Selection |
| && | **X** |
| == |  |
| || |  |
| != |  |

***(1 Mark)***

**Question D**

|  |
| --- |
| In the space provided below, declare a string variable called **name** and initialise it to **Bailey**.  *Take care with capitalisation* |
| string name = “Bailey”; |

***(2 Marks)***

**Question E**

This question is divided into 3 parts.

|  |  |
| --- | --- |
| **1.Which of the following keywords is used to pass a value back from a function**  Please choose one of the following options by entering X into the box of your selection | |
| Option | Selection |
| for |  |
| return | **X** |
| pass |  |
| int |  |

***(1 Mark)***

|  |
| --- |
| **2. In the space below, write a function called addTax that adds the VAT to a price of an item.**  Your function should take the original price as a parameter and return the price with the tax added.  You should use the following calculation to calculate the price with VAT added:  **originalPrice \* 1.2**  You do not have to write any code to call the function from int main(). |
| void addTax(float originalPrice)  {  originalPrice\*=1.2;  return originalPrice;  } |

***(3 Marks)***

|  |
| --- |
| **3. State an advantage of using functions within your code.** |
| Functions provide an effective readability and maintainability of code. |

***(1 Marks)***

# Documentation

## Variables

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Type** | **Description** |
| Option\_Selector | int | Option\_Selector will be used to select option given by the program. |
| Credit | double | Credit is a variable that will store Credit. |
| Pizza\_Price | double | Pizza\_Price variable is used to assign the Pizza Prices depending upon the size of Pizza. |
| Topping\_Price | double | Topping\_Price variable is used to assign the Topping Prices depending upon the chosen Topping. |
| Total\_Price | double | Total\_Price is a variable that stores a sum of Pizza & Topping Prices.  (Total\_Price = Pizza\_Price + Topping\_Price;) |
| Additional\_Credit | double | Additional\_Credit is a variable that stores credit. Which is to added to “Credit”. |
| Required\_Balance | double | Required\_Balance is a variable that has Credit needed to process the order. |
| yesno | string | yesno is a variable that will used to store the response of user to any question throughout the program. |

## Methods & Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | yesno\_Verify | **Return Type** | void |
| **Parameter** |  | | |
| **Description** | yesno\_Validation() is a function, which will be used for the taking the input of “**yesno**” variable. It will be also used for validating user’s input and will display an error message if inputs are wrong and would ask for input again until its right input. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Credit\_Validation | **Return Type** | void |
| **Parameter** |  | | |
| **Description** | Credit\_Validation() is a function used to validate the user’s input for Credit. I will ensure the Credit is between 1-999 and is numbered. It will show an error message, if input is wrong and ask for input again until Credit input is right. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Checkout() | **Return Type** | void |
| **Parameter** |  | | |
| **Description** | In the “**Checkout**” function, it is supposed calculate and checkout for the order.  Checkout function is going to have two “**if statement**” for the whether the “**Credit**” is enough to checkout or not. Both will have their own outcomes to ensure that the user has enough money to purchase an order.   * If “**Credit**” is more than or equal to “**Total\_Cost**” for the checkout. Program will continue and simply checkout list: * Available Balance. * Pizza Price * Topping Price * Total Cost * Credit to be taken by Customer. * If there isn’t enough “**Credit**” for the checkout. Program will ask for whether the user wants to add some “**Additional Credit**” and continue with the processing order or not. * If user wants to continue with Processing order, it will Checkout and give remaining Credit to the user. * If user doesn’t want to continue with the Processing order, it will be returned the whole amount of Credit.   Overall, “**Checkout**” function would perform calculation as well as ensure that the balance is enough to be able to make a purchase of an order. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | main() | **Return Type** | int |
| **Parameter** |  | | |
| **Description** | main() is a function that will be used for plenty of purposes. It will be used for the outputting “**Main Menu**”, it would also have user input validation checks. “Main Menu” would have three options:   * 1. Add Credit * 2. Order Pizza * 3. Exit   Mainly, this function consists three “**If** **statements**” :   1. If user chooses to “**Add Credit**”, it can add Credit and the function would return to “**Main Menu**” again. It would also run “**Credit\_Validation()”** to validate the users input. 2. If user chooses “Order Pizza”, it will be asked for pizza and topping while running validation checks for users input. Ultimately, it will run “Checkout()” function and show remaining balance as well final “Good Bye” message. 3. If user chooses “**Exit**”, then the program will shut down. | | |

## User Guide

Program consists of mainly three option to make it easy and efficient for users. Program will allow to choose one of the options from:

**Shape

Description automatically generated with medium confidence**

If user chooses “**1. Add Credit**”, it will be allowed to enter the “**Credit**”, once “**Credit**” has been added. Program will once again return to “**Main Menu**”. So, User can continue with the rest of program.

Text

Description automatically generated

If user chooses “**2. Order Pizza**”, it will allow the user to choose Pizzas and its Toppings as well as additional toppings if wanted.

* Once the Pizza and Topping has been selected by the user. Program will go ahead to “**Checkout**”.
* If **“Credit”** is enough to pay for the transaction, which user is trying to make. Program will go ahead and “**Checkout**” the order and ask for whether user wants to order another pizza.

Text

Description automatically generated

* If Credit isn’t enough to pay for the transaction. Then, the program will show the **insufficient** **credit** needed to continue transaction.

A screenshot of a computer

Description automatically generated with medium confidence

The program will ask whether the user wants to add more “**Additional Credit**” to continue the transaction and it will keep asking for the “**Credit**” till it is more than or equal to “**Total\_Cost**”. Otherwise, if user doesn’t want to add the credit and continue the transaction, it will be given all “**Credit**” back and it will asked again, if the user wants make another order for the pizza.

* Once the “**Credit**” is enough to continue transaction. Program will ask for whether the user wants to “**Continue Process**”. If “**Yes**”, then the program will continue. If “**No**”, then the program will give “**Remaining Balance**” to the user.

**Text

Description automatically generated**

* Ultimately, after the “**Checkout**”, the program will give an option to order another pizza.

If user chooses “**0. Exit**”, it will exist the program.

Overall, the program would validate all the user’s inputs. Therefore, if there is any wrong input being entered, the program would show an error message and ask for the input again and it will keep repeating until the right input has been entered.

# Testing

The following use-case will be used to test aspects of your application. Please provide sufficient evidence (this includes screenshots of your application) to demonstrate how your application fulfils the testing use-case criteria defined below.

## Use-case

|  |
| --- |
| 1. Add **5.00** credits to your balance.   Show before and after screenshots |
| **Before:**  **Shape  Description automatically generated with medium confidence**  **After:**  Text  Description automatically generated |
| 1. Select a **7-inch** pizza.   Show before and after screenshots |
| **Before:**  **Text  Description automatically generated**  **After:**  Text  Description automatically generated |
| 1. Add a **Ham** to your pizza.   Show before and after screenshots |
| **Before:**  **Text  Description automatically generated with medium confidence**  **After:**  Text  Description automatically generated |
| 1. Add a **Pepperoni** to your pizza.   Show before and after screenshots |
| **Before:**  **Text  Description automatically generated**  **After:**  Text  Description automatically generated |
| 1. Add a **Mushrooms** to your pizza.   Show before and after screenshots |
| **Before:**  **Text  Description automatically generated**  **After:**  Text  Description automatically generated |
| 1. Attempt to **Checkout**   Show before and after screenshots |
| **Before:**  Text  Description automatically generated  **After:**  **A screenshot of a computer  Description automatically generated with medium confidence** |
| 1. Add **2.00** additional credits to the vending machine   Show before and after screenshots |
| **Before:**  **A black screen with white text  Description automatically generated with low confidence**  **After:**  Text  Description automatically generated |
| 1. **Checkout**   Show before and after screenshots |
| **Before:**  **Text  Description automatically generated**  **After:**  **Text  Description automatically generated** |
| 1. **Show remaining balance**   Show before and after screenshots |
| **Before:**  **Text  Description automatically generated**  **After:**  **Shape  Description automatically generated with medium confidence** |

# Reflection

There were plenty of the challenges faced throughout the assignment that would genuinely have an impact on students’ abilities in terms of approaching programming tasks. One of the main things was that the programs requirements or main feature & functionalities must be written down to have an idea what kind of program needs to be developed. There should also flowchart of the program.

Another extremely important take away from this would be to fixing bugs efficiently without affecting the rest of the functionalities of the program. Otherwise, one bug will be solved, and another would appear.

There were also a few challenges with the usage of while loops, do while loops and if statement. Even though, they were easier to use. The complexity and mixture as well the structure of the program was making it hard to implement some of the features and functionalities.

Useful resources provided on the blackboard were very helpful to overcome different challenges. However, there were also a few challenges that were overcome by trying and testing different loops and statements.

Fulfilling the requirements of the assignment’s tasks may not be considered as extremely hard. However, bearing in mind that the code should be as condense as it can be to have good readability and maintainability is what would be considered as hard. To fight this challenge, my approach was to write a program that meets all the asked requirements and gradually condense with the usage of effective and right code. Ultimately, I intended to add informative comments to explain my program to improve of my readability.

Overall, the assignment has provided me with the good-knowledge and skills of problem solving as well managing my programs feature & functionalities consistent throughout the fixing bugs phases.