

**School of Psychology and Computer Science**

**CO1404 – Introduction to Programming 2020/2021**

* **Assessment 1**: Coursework
* **Title:** UCLAN Vending Machines LTD

**Name:** <Adrian Mazur>

**Email :** <amazur@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

UClan Vending Machine Assignment Report Intro

Students have been tasked with the development of a new program for the vending machines at the university. They must develop a console-based vending machine program that demonstrates the vending machines functionality to a group of potential investors. THE list of items, functions and features is already defined to the students and they must stick to them, they can however add more features as they see fit. It is also important that the students work on this individually and not in groups.

As well as the program they have to make a User Guide and must write up a reflection of how they feel about themselves and the skills they used/ had to develop while developing this program. The students also need to show proof (by using screenshots of their program in action) to test the program and show how it meets the required criteria asked for.

# 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#, what key word is used to return a value from a method?  Please choose one of the following options by entering **X** into the box of your selection | |
| Options | Selection |
| return |  |
| get |  |
| if | **X** |
| for |  |

***(1 Mark)***

**Question B**

|  |  |
| --- | --- |
| Which of these is not a reason to improve the readability of code?  Please choose one of the following options by entering X into the box of your selection | |
| Option | Selection |
| It will help other members in a programming team. |  |
| It will improve program efficiency. | X |
| It will help you understand your own code in the future. |  |
| It will make your program structure clearer. |  |

***(1 Mark)***

**Question C**

|  |  |
| --- | --- |
| ***What is the value of result after executing this code?***  int result; int value = 20; if (value > 0 && value <= 10) {  result = 1; } else if (value > 10 && value <= 20) {  result = 2; } else if (value > 20 && value < 30) {  result = 3; }  Please choose one of the following options by entering **X** into the box of your selection | |
| Options | Selection |
| The code cannot be executed because it contains an error. | **X** |
| The value of Result is undefined. |  |
| 2 |  |
| 3 |  |

***(1 Mark)***

**Question D**

|  |
| --- |
| ***In the space provided below, declare an integer variable called cost and initialise it to 25.***  *Take care with capitalisation* |
| int cost = 25; |

***(2 Marks)***

**Question E**

This question is divided into 3 parts. Part 1 is worth 1 mark, Parts 2 and 3 are worth 2.

|  |  |
| --- | --- |
| ***1. Which of the following is not an example of a loop in C#?***  Please choose one of the following options by entering X into the box of your selection | |
| Option | Selection |
| For |  |
| While |  |
| Do While |  |
| Nesting | **X** |

***(1 Mark)***

|  |  |
| --- | --- |
| ***2. Write a nested for loop which will prints each number individually from 0 to 9, on a single line 3 times as shown below.***   |  | | --- | | 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9  0 1 2 3 4 5 6 7 8 9 | |
| int counter = 0;  for (int i = 0; i < 1; i++)  {  for (int j = 0; j < 10; j++)  {  Console.Write("{0} ", counter);  counter++;  }  Console.WriteLine();  Console.WriteLine();  int counter1 = 0;  for (int j = 0; j < 10; j++)  {  Console.Write("{0} ", counter1);  counter1++;  }  Console.WriteLine();  Console.WriteLine();    int counter2 = 0;  for (int j = 0; j < 10; j++)  {  Console.Write("{0} ", counter2);  counter2++;  }  Console.WriteLine();  Console.WriteLine();  }  Console.ReadLine(); |

***(2 Marks)***

|  |  |
| --- | --- |
| ***3. Calculate the number of iterations that is required to complete the code above and explain your answer.*** | |
| Number of iterations | **30** |
| Explanation:  The number of iterations must be 3 for the order of numbers to be printed 3 times, otherwise if the code was repeated less than 3 times then the numbers wouldn’t be printed 3 times but less. | |

***(2 Marks)***

# Documentation

## Variables

|  |  |  |
| --- | --- | --- |
| **Variable Name** | **Type** | **Description** |
| item1 | string | A char that is set to the name of the 1st food choice in the vending machine |
| item2 | string | A char that is set to the name of the 2nd food choice in the vending machine |
| item3 | string | A char that is set to the name of the 3rd food choice in the vending machine |
| item4 | string | A char that is set to the name of the 4th food choice in the vending machine |
| item5 | string | A char that is set to the name of the 5th food choice in the vending machine |
| item1Price | double | The double data type is used to display the price of item 1 in decimals |
| item2Price | double | The double data type is used to display the price of item 2 in decimals |
| item3Price | double | The double data type is used to display the price of item 3 in decimals |
| item4Price | double | The double data type is used to display the price of item 4 in decimals |
| item5Price | double | The double data type is used to display the price of item 5 in decimals |
| credits | double | Displays the credits entered by the end user as decimals |
| validCredits | bool | Checks whether credits entered by the end user are valid (whether they are numbers) and returns a true or false statement |
| yesNo | string | Stores user’s response, either a yes or a no |
| isNumber | bool | Checks whether the response pressed by the user is a number or not |
| vendingChoice; | int | Stores the user input while they are in the food selection option |
| mainMenuInput | string | Stores the end users input while they are in the Main Menu and enables them to input their response by clicking the numbers corresponding to the available choices displayed at the time |
| subTotal | double | Used to calculate the sub total price of the end users choice/s and display it as decimals |
| finalCheckout | string | Used to ask the end user whether they want to go to the final checkout by asking them to enter y for yes or n for no |
| finalCredit | double | Used to display the final credit amount of the end user in the final checkout stage as decimals |
| Item01Price | double | For testing purposes. This is similar to item1price but needed a new name to work for the checkout feature |
| Item02Price | double | For testing purposes. This is similar to item2price but needed a new name to work for the checkout feature |
| totalCost2 | double | For testing purposes. This is variable calculates the total cost of items 2 and 3 and displays result in decimals |

## Methods & Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | break | **Return Type** | void |
| **Parameter** | void | | |
| **Description** | When it is used inside a loop, this loop will then be terminated, and the program control will resume at the next statement following the loop. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | return | **Return Type** | void |
| **Parameter** | void | | |
| **Description** | It terminates an execution of a method in which it appears in and returns control back to the method that was used to call it. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | TryParse | **Return Type** | Boolean value |
| **Parameter** | Out | | |
| **Description** | TryParse will return a bool value of true or false. If it works, it will return true however if it will not work it will return false. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | Do While loop | **Return Type** | Boolean value |
| **Parameter** | Boolean value | | |
| **Description** | The Do block will be executed at least once and will loop until the condition is met. And the condition in the while part will be used to evaluate a specific variable after the do block has been executed. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | ToLower | **Return Type** | Converted string from uppercase to lowercase |
| **Parameter** | Converted string from uppercase to lowercase | | |
| **Description** | This is a String method which will convert each character which was typed in by the user to lowercase. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | logical AND operator (&&) | **Return Type** | Boolean value |
| **Parameter** | Condition | | |
| **Description** | It returns true if both or all the conditions are true and return false if any of the condition is false. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | DrawHorizontalLine | **Return Type** |  |
| **Parameter** | Int lineWidth | | |
| **Description** | Draws a horizontal line of width 80 across the display screen by calculating the empty spaces | | |

## User Guide

Welcome to the user guide for our vending machine. Please read it carefully to understand how it works, pictures are attached for further help. Please be aware that we are continually working on updating it and will be updating this guide with every new change.

Welcome Screen

Below is what the welcome screen looks like once you launch the vending machine. **Text

Description automatically generated**

Main Menu

Below is what the Main Menu looks like. To see it, press the “Enter” key.

Here the end-user is presented with 4 choices:

1. Add credits to make a purchase.
2. Display the available products alongside with their prices.
3. And an Exit option to quit the vending machine.
4. Checkout option to make the final payment.

After these options are displayed the user must press either of the following keys 1, 2, 3 or 0 to make a choice. Press either of the specified numbers followed by ENTER.

Text

Description automatically generated

Displaying Items

If key 2 is pressed followed by ENTER all the items and their subsequent prices will be displayed.

Text

Description automatically generated

Choosing an item

After pressing ”ENTER” a message prompting the user will be displayed, press a number from 1 to 5 to choose what function to call next.

In the picture below is what each key brings up, e.g. If key number 1 followed by ENTER is pressed the following message will be displayed “You have chosen Chocolate Bar, it costs 0.8 credits (which is 0.80 credits)”.

Text

Description automatically generated

Going back to the Main Menu

If the user wishes to go back to the main menu then after the message “Do you want to add to your order?” press Y followed by ENTER and then after the message “Please enter a number from 0 to 5” Press 0 and then ENTER.

Text

Description automatically generated

Quitting the Vending Machine

After pressing 0 and ENTER the following text will be displayed like in the picture below.

Text

Description automatically generated

Text

Description automatically generated

Adding Credits

After pressing “ENTER” the main menu will show up. Below the 4 options “Add credits, Display product/s, Checkout and Exit” a message prompting the user to press either 0, 1, 2 or 3 followed by ENTER will pop up. To add credits, the user needs to press key 1 and ENTER, after a message “input credits” will show. This is where the user inputs the credits and then must press ENTER, this will then display how much credits have been added.

Text

Description automatically generated

Checkout

From the main menu, press 3 once prompted to go to the checkout (this checkout feature is a future feature and is not fully functional). After pressing 3 a message “Welcome to the checkout” will be displayed. Press ENTER and this will end the program (a future update will be put in place to make sure this feature will be fully functional).

Text

Description automatically generated

How to check out the right way

After finishing adding to the order press ‘n’ when prompted if you want to add to your order, then press ENTER 2 times.

Next the following prompt will be displayed “Do you want to proceed to checkout?” Press ‘y’ to see the grand total, how many credits there are and the final balance.

The checkout feature is not fully functional as it only calculates the grand total of items 2 and 3.

Text

Description automatically generated

Text

Description automatically generated

Error messages & their solutions

Item selection error & Solution

If during the item selection stage, you press something other than the numbers, 1,2,3,4,5 or 0 a “enter valid number” message will be displayed and will keep being displayed until either of the above numbers will be pressed.

Text

Description automatically generated

Adding credits error & Solution

Text

Description automatically generated

If when adding credits, the user enters something that is not a number a prompt will appear “input credits”. The solution is to enter numbers and the error message will disappear and the program will let you move on to item selection.

# 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

|  |
| --- |
| * Add **3.50** credits into the vending machine.   Show before and after screenshots |
| Text  Description automatically generated  Text  Description automatically generated |
| * Add a **Soda** **Can** to the **order**   Show before and after screenshots |
| A picture containing calendar  Description automatically generated  Text  Description automatically generated |
| * Add a **Cookies** to the **order**   Show before and after screenshots |
| Text  Description automatically generated  Text  Description automatically generated |
| * Add a **Soda Can** to the **order**   Show before and after screenshots |
| Text  Description automatically generated  Text  Description automatically generated |
| * Add a **Soda Bottle** to the **order**   Show before and after screenshots |
| Text  Description automatically generated  Text  Description automatically generated |
| * Attempt to **Checkout.**   Show before and after screenshots |
| Text  Description automatically generated  Text  Description automatically generated |
| * Add **5.00** additional credits to the vending machine.   Show before and after screenshots |
|  |
| * **Checkout**   Show before and after screenshots |
|  |
| * **Show remaining balance.**   Show before and after screenshots |
|  |

# Reflection

# Accomplishments

* Managed to get the program to read and accept the number of credits that the end user has inserted.
* Credits inserted now show up but not all the way throughout the program.
* Food & Drink option are now displayed at the start by the use of strings (to hold and display the options to choose) and doubles (to hold and display the prices in decimals).
* Choosing an item now displays its price as well.
* Option to add additional items to your order is available by prompting the user with a yes or no question, if yes, they can then choose a number of the item to add it to the order. If not, the end user can go to the checkout which is done by the use of a new Do While loop
* Checkout feature has now been implemented.
* Subtotal feature implemented although not fully functional.
* Overhaul of the user interface by the addition of methods (DrawText) which make the title texts be displayed in the middle.
* In addition to the user interface, I have managed to also add a DrawHorizontalLine method which draws a line of stars that helps to make the program look more aesthetically pleasing to the end users.

Challenges

* Displaying total cost after user finishes choosing items
* Storing user input
* Displaying the right item
* Displaying price next to chosen item (This has been achieved by declaring some double variables and assigning names and values to them and then adding them together to display a sub total value)
* Showing a grand or sub total price
* Error detection/ catching.
* Implementing methods but that was no longer a problem after looking over a previous lab sheet about draw text methods. However, there are more methods that I would have liked to add that would help with outputting the credits that are left after the completion of the checkout and calculating the total credits throughout the program.
* Using Arrays correctly (I have managed to use an Array, but it only displayed the prices of the items and I could not figure out how to make the Array also show up the name of the item that the end-user has chosen)

Improvements

* Reduce amount of code so the program looks tidier and is more readable.
* Include comments so the code is easier to understand, this will also make it easier to make amendments to the code and even improvements to it.
* After user inserts credits amend the code so that the user does not have to guess what each number will end up selecting, in other words make it show a table of items and their prizes again. (This has now been updated as of 27/12/2020)

Short term goals

* Add code for calculating total cost so a transaction can be completed by the user.
* Add code that displays to the user how much of the specific item is left and if it is in stock.
* Add a checkout feature so the end user can see total cost and then decide whether they want to pay or cancel their order (Currently the checkout feature is not fully operational)
* Add code to ask the user how many of the item they wish to add to their order.
* Add code to calculate credits currently in the vending machine.
* Add code to display to the user how much change they will be getting.
* Continue development to improve functionality and usability of the Vending Machine so that the end users are satisfied with it.
* Once the checkout feature is fully implemented, the application should check if the number of credits is sufficient to complete the end user’s transaction and if not, it should prompt the end user to insert more credits.

Long term goals

* Have a good system maintenance and security in place to prevent hacking attempts, make sure to gain knowledge of good security and have security experts that know how to deal with this.
* Implement scheduled security patches and updates to keep up with any changes/ amendments in terms of the security and any legislations in place.
* Provide an updated User Guide to keep up with any changes/ alterations to the User Interface so that the users can still know how to use the Vending Machine.
* Continue development to improve functionality and usability of the Vending Machine so that the end users are satisfied with it.
* Continue developing my programming skills so that I am more confident when writing and developing my own programmes in the future.

Conclusion

This was an interesting project to work on and there is a lot I learnt from it, I also got to see where I am lacking knowledge and/ or experience in.

This was the 1st project I ever attempted which gave me a chance to put all my learning so far into practice. And coming from a non-IT background, undertaking such a project was a new and unknown venture for me but nevertheless an exciting one.

I have in the end managed to develop a program with reasonable success. Although I was not able to fully implement all 3 of the required features, they all were lacking something to make them fully functional. However, I am sure that with time I will have enough knowledge and experience to develop this program further and even implement some further features to make the program even better.