

Lloyd Case 1818917

by Lloyd Case

Submission date: 10-Jan-2024 01:33PM (UTC+0000)

Submission ID: 221362460

File name: 23995_Lloyd_Case_Lloyd_Case_1818917_532657_816891567.docx (118.65K)

Word count: 1559

Character count: 7089

ACCB5011
ADVANCED SOFTWARE
DEVELOPMENT
OOP PROGRAMMING (C++)

Lloyd Case (1818917)

Test Plan

In this test plan I will be validating all the functionalities of the application as well as having certain test scenarios that will be used to see what outcomes are expected from each different function. I will be testing to see if everything is working correctly when looking at the things that need testing all the user input will need testing to see if everything is working correctly and nothing goes wrong.

```
Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5: 1
Enter name of product: Apple
```

Firstly, I will be testing the “Add a product” feature that is in my program. This is supposed to give the user the option to add a product which can be named anything as it is user input. As seen in image number 1, this is working as predicted and is working correctly as the user was able to input the product without any issue.

```
Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5: 1
Enter name of product: Apple
Enter product quantity: 10
The Product has been added to the inventory.
```

Now as seen in the second image we were able to give the product a number as in the quantity of the product that is stored in the inventory.

The Product has been added to the inventory.

This is noted by the caption underneath stating that the “This product has been added to the inventory”, now we can check to see if this product has been added to the inventory by using number 4, which is to display the current inventory.

```
Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5: 4
Currently in the inventory
The product: Apple
Quantity of product: 10
```

As seen in this image it is displayed that the user input the number 4 to see the current inventory, and the application has displayed the current inventory which is the product and the quantity of the product. This is working as expected with no issues to note.

```
Please enter a number between 1-5: 1
Enter name of product: Banana
Enter product quantity: 5
The Product has been added to the inventory.
```

And in this image, we can see that the user is able to add another product to the inventory with no issues again as well as a different number in the quantity section to show that the application can hold more than just the one number.

```
Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5: 4
Currently in the inventory
The product: Apple
Quantity of product: 10
The product: Banana
Quantity of product: 5
```

Then shown in this image we can see that the display inventory is again working correctly and is now displaying both products as well as the quantity of both products. So, the application can keep such data while getting new data from the user inputs.

```
Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5: 2
Enter name of product to remove: Apple
Product successfully removed from the inventory.
```

In this image it is shown that the inventory can remove products, in this image it is shown that the “Apple” product has been successfully removed from the inventory by showing the statement “Product successfully removed from the inventory.”

```
Currently in the inventory
The product: Banana
Quantity of product: 5
```

Then shown in this image we can see that the “Apple” product has been successfully removed as stated before and now the inventory now only displays the “Banana” product that was added afterwards.

```
Enter name of product: Banana
Enter updated quantity of product: 10
Product quantity updated.
```

In this image it is shown that the quantity of the product is getting increased by the updated number. It was previously 5 and this is increasing it by 10.

```
Currently in the inventory
The product: Banana
Quantity of product: 15
```

As shown in this image we can see that it has been increased by 10 and is now displaying 15 as that is the updated number.

```
Currently in the inventory
The product: Banana
Quantity of product: 15
The product: Apple
Quantity of product: 10
The product: Orange
Quantity of product: 12
The product: Pear
Quantity of product: 45
The product: TV
Quantity of product: 8
The product: PC
Quantity of product: 23
The product: Bed
Quantity of product: 4
```

Shown here are multiple products added to the inventory and still working correctly, by displaying all the products with their own different quantities. As it is all users input these products as well as quantities can be whatever the user inputs.

```
Enter name of product: tv
Enter updated quantity of product: 12
Product not currently in the inventory.
```

The program still working correctly here is showing that if you try to add quantities to a product that does not exist, the application will tell you so by stating “Product not currently in the inventory”. As you can also see in the image, the

product is named “tv”. The reason this does not show up in the inventory is because it is case sensitive, meaning you will have to remember to capitalize the letters before inputting or remember what has been capitalised as the inventory will not recognise what the product is unless the product is “TV” not “tv”.

```
Please enter a number between 1-5: 1
Enter name of product: Apple
Enter product quantity: 1
The quantity of the product has been updated.
The product: Apple
Quantity of product: 11
```

In this case we can see that when trying to add a product that already exist will seem like its prompting to make another product of the same name, however when you enter the quantity while it does ask the same question, when looking at the last line you can see that it has actually added the quantity to the already existing product which in this case is the “Apple”, when displaying our inventory we can now see that it has indeed added the one to the “Apple”.

```
Please enter a number between 1-5: 2
Enter name of product to remove: tv
This product is not currently in the inventory.
```

As shown in this image we can see that when trying to remove a product that does not exist the application will tell you that there is no product with that name in the inventory therefore will not remove anything.

```
Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5: 5
Closing Inventory
```

Then when entering the number 5 to close the inventory you are then prompted with “Closing inventory”, then the inventory will close, closing the application.

Errors

In this section I will be showing what errors can occur to break the program. These will not occur if you follow the instructions on what to do but can however happen by mistake and when an error occurs you will have to close the application.

```
Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5:
Invalid choice. Please enter a valid option: 1, 2, 3, 4, 5

Inventory Management System
1.Add a product
2.Remove a product
3.Update the quantity of a product
4.Display inventory
5.Close inventory
Please enter a number between 1-5:
Invalid choice. Please enter a valid option: 1, 2, 3, 4, 5
```

This is difficult to show in the image however if you input a letter instead of a number when selecting a choice, this will cause the application to keep repeating the invalid choice statement. Because of this you will be forced to close the application. This is not the only error in the application. This error occurs anytime you input a letter where a number must be the input.

Importance of OOP concepts

What is OOP? OOP consists of these structures: Classes, Objects, Methods, Attributes [1]. Classes are the blueprint for the objects that will be used within the class, the objects go with it [1]. Objects are what go in the class, such as the details of the class, more in depth of what it is [1]. The methods are the function in the class that is the behaviour of the objects. Attributes are what the data is stored in, it's what the class itself is [1].

Object-Oriented Programming (OOP) is important to creating an optimised program using C++. OOP is very advantages as it is faster and easier to execute [2], makes the code clear and easier to keep track of what you are doing, makes it so you don't have to repeat the same code over and over and because of less code this does mean that it is quicker and simpler to make applications [2].

However, when using OOP, it must be noted that while the code is more clear and easy to understand, it will become a bigger program, meaning there will be much more lines of code to write in the program itself as you will be using classes to call certain pieces of code to use at specific times

[3]. OOP is not easy to use and will take a lot of time to get used to using it. The reason for this is that it requires a lot of knowledge on when and how to use it effectively [3]. As it is OOP every time you use it, what you use it for will make that into an object, meaning that you must think of it like and object no matter what you are using it for [3].

The way that I used OOP concepts in the application were as follows. Created the IMS class which is used to allow for the ease of bringing everything in the IMS class to the front using the int main (). By doing this I can clearly see what is being used in the program when using the IMS class, using this class means that the int main () isn't getting obstructed with a lot of code that makes it unclear what is being used. I put all the code needed for the IMS in the IMS class and then call the IMS class forward into the main program. There are the lines of code that use the void xyz(code), using these it is easy to see the use of OOP because I can call them whenever I want to in the code instead of having to re write those lines every time, I would want to use it.

There is a lot of flexibility when using OOP in your coding. There are many reasons why OOP is helpful to the programmer and is worth learning as it will cut down the time it takes to code as using classes helps decrease the amount of code as everything for a certain subject can be stored within the class, such as the objects you will be using because of the class [4]. It is easier to troubleshoot

when the code isn't working as you will know where to look because there is only one instance of the code in the project  so there will be less to look through if something goes wrong [4], this also helps  debugging as the code will be clear on what is written and this will help with the productivity when writing the code [4].

References

- [1] A. S. Gillis, "object-oriented programming (OOP)," Tech Target, July 2021. [Online]. Available: <https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP>. [Accessed 10 January 2024].
- [2] "C++ OOP," w3schools, [Online]. Available: https://www.w3schools.com/cpp/cpp_oop.asp. [Accessed 10 January 2024].
- [3] "Advantages and Disadvantages of OOP," Geeks For Geeks, 2 March 2022. [Online]. Available: <https://www.geeksforgeeks.org/benefits-advantages-of-oop/>. [Accessed 10 January 2024].
- [4] G. Aggarwal, "Benefits Of OOP In C++: All You Need To Know," Intern Shala, 12 June 2023. [Online]. Available: <https://trainings.internshala.com/blog/benefits-of-oop-in-cpp/>. [Accessed 10 January 2024].

Lloyd Case 1818917

GRADEMARK REPORT

FINAL GRADE

58/100

GENERAL COMMENTS

Hi Lloyd,

In-depth feedback is provided within the document, and a summary is presented here. Scores for individual sections are accessible through the rubric.

Areas of appreciation:

Your test plan provides a comprehensive overview of the functional testing performed on the Inventory Management System (IMS) application. It's commendable that you've included detailed descriptions and images to illustrate the outcomes

Areas of improvement:

Explicitly include negative testing scenarios

OOPs features should have been explained in detail

Code uploaded in the additional file link was not an executable file.

I hope these comments are useful

PAGE 1

PAGE 2



Comment 1

A brief introduction about the chosen application system and functions to be included could have been a good start



Comment 2

The output statement acknowledging the successful execution is well done.

PAGE 3

PAGE 4



Comment 3

proper error messages to let the user know about the error could have been the best option instead of closing the application



Comment 4

we can start using abbreviations only when it has been already mentioned



Comment 5

this statement contradicts with one of the oops features



Comment 6

the code snippets showing the implementation of oops feature could have been included.

PAGE 5



Comment 7

explanation about the code should also be included



Comment 8

A minimum of 3 features of oops could have been explained



Comment 9

References should always be from creditable sources.

GRADING FORM: DC-ASD

LLOYD CASE

58

LITERATURE REVIEW

with IEEE referencing



3

KNOWLEDGE&UNDERSTANDING SKILLS

use of OOPs features



7

COGNITIVE&INTELLUCTUAL SKILLS

Testing &Error handling



11

PRACTICAL SKILLS

Working program Demo video



9

TRANSFERRABLE SKILLS

User Manual, document style



3

CISCO CERTIFICATE



25