



slington college
(इस्लिङ्टन कलेज)

Fundamental of Computing

60% Individual Coursework

2023 Spring

Student Name: Riza Shrestha

London Met ID: 22067117

College ID: np01nt4a220087

Assignment Due Date: Friday, 28 July 2023

Assignment Submission Date: Friday, 28 July 2023

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

Tables of Figures	3
Tables of tables	4
1. Introduction.....	1
1.1 Introduction of the project.....	1
1.2 Goal and Objectives.....	1
2. Algorithm.....	3
3. Flowchart	5
4. Pseudocode	6
4.1 For main.....	6
4.2 For operations file.....	11
4.3 For Read file.....	15
4.4 For Write File.....	17
5. Data Structure.....	23
5.1 Primitive data type	23
5.2 Non-primitive data type.....	24
6. Program	26
7. Testing.....	32
7.1 Test 1: Show implementation of try, except.....	32
7.2 Test 2: Selection purchase and sale of laptops	34
7.3 Test 3: File generation for laptop purchase	36
7.4 Test 4: File generation for laptops sales process	40
7.5 Test 5: Show the update in stock of laptops.....	44
8. Conclusion.....	49
9. Appendix.....	50
9.1 For Main.....	50
9.2 For operations.....	56
9.3 For read	59
9.4 For write.....	61

Tables of Figures

Figure 1: Flowchart of the program.	5
Figure 2: Integer data structure of the program.....	23
Figure 3: String data structure of the program.....	23
Figure 4: List data structure of the program.....	24
Figure 5: Dictionary data structure of the program.	25
Figure 6: Screenshot of the program's format.	26
Figure 7: Screenshot of Sale laptop.....	27
Figure 8: Screenshot of purchase laptop.	28
Figure 9: Screenshot of exiting the program.....	29
Figure 10: Screenshot of before purchase and sale.....	29
Figure 11: Screenshot of quantity of laptop after selling.	29
Figure 12: Screenshot of quantity of laptop after purchasing.....	30
Figure 13: Screenshot of the file before purchase and sale.....	30
Figure 14: Screenshot of file after purchase and sale.....	30
Figure 15: Screenshot of after purchase in text file	31
Figure 16: Screenshot of after sell in text file.....	31
Figure 17: Screenshot from Test 1	33
Figure 18: Screenshot from Test 2 - Negative value.....	35
Figure 19: Screenshot from Test 2 - Non-existing value	35
Figure 20: Screenshot from Test 3 - Showing the laptop's complete purchase process and output.	38
Figure 21: Screenshot from Test 3 - Information about purchased laptops in a text file.....	39
Figure 22: Screenshot from Test 4 - Showing the laptop's complete sale process and output. ..	42
Figure 23: Screenshot of Test 4 - Information about sold laptops in a text file.....	43
Figure 24: Screenshot from Test 5 - Showing the laptop's complete purchase and output.	45
Figure 25: Screenshot from Test 5 - Information about purchased laptops in a text file.....	46
Figure 26: Screenshot from Test 5 - The amount added after purchasing the laptop.	46
Figure 27: Screenshot from Test 5 – showing the laptop's complete sale process and output. ..	47
Figure 28: Screenshot from Test 5 - Information about sold laptops in a text file.	48
Figure 29: Screenshot from Test 5 - The amount deducted after selling the laptop.	48

Tables of tables

Table 1: Test 1 - Verify the functionality of try- except block.....	32
Table 2: Test 2 - Verify the program's handling of negative value and non-existing value as input.	34
Table 3: Test 3 - Verification of laptop purchase file generation.....	36
Table 5: Test 5- Verifying the update in stock of laptops.	44

1. Introduction

1.1 Introduction of the project

In this project, a system was created for a laptop store that buys laptops from manufacturers and resells them to different customers. The store also accepts orders for computers from customers and manufacturers. The system uses a text file to hold the information about the laptops, and the program is made to read the file, display the laptops that are available, and make the appropriate changes to the file based on the type of transaction, such as ordering from manufacturers or selling to consumers. The program also creates receipts for each transaction and provides detailed information about the transaction, including the specifics of the laptop, the customer, the total amount, the shipping cost (for sales), the VAT amount (for orders), and the total amount to be paid.

The laptop store can manage its inventory, fulfill orders, create transaction records, assure accurate stock updates, and more with the help of this system, ultimately improving operational effectiveness and customer service.

1.2 Goal and Objectives

The goal of this project is to give students the opportunity to develop an understanding of databases and obtain practical learning experience while concentrating on designing and implementing an application for a laptop store. The main objective of this project is:

- To develop a system to efficiently manage the store's laptop inventory that will make it simple to keep track of the computers that are available, their availability, and other related data.
- To create a program that enables the laptop store to effectively handle sales transactions with customers and to improve the ordering process between the laptop shop and manufacturers, ensuring accurate record-keeping and easy communication.

- To improve customer satisfaction with better customer service through accurate stock availability, production of complete transaction records, and enabling of smooth an error-free transactions.

2. Algorithm

STEP 1- START

STEP 2- DISPLAY the available options for the user

STEP 3- IF user input is 1, go to STEP 4, else if go to STEP 14, else go to STEP 26

STEP 4- DISPLAY message to prompt the user to provide customer details (name and phone number)

STEP 5- INPUT the customer details (name and phone number)

STEP 6- DISPLAY the details of the available laptop stock

STEP 7- INPUT the id of the laptop to sell

STEP 8- IF id of the laptop is less than or equals 0 or greater than 5, go to STEP 7, ELSE go to STEP 9

STEP 9- INPUT the quantity of laptops to be sold

STEP 10- IF the quantity of the laptop is zero, go to step 9, else go to step 11

STEP 11- DISPLAY message to ask if the user wants to continue selling (y/n)

STEP 12- IF user input is "y", go to STEP 7, else go to STEP 13

STEP 13 - Generate the purchase receipt and go to step 2

STEP 14- IF user input is 2, go to step 15, else if go to step 4, else go to STEP 26

STEP 15- DISPLAY message to prompt the user to provide manufacturer details (name and phone number)

STEP 16- INPUT the manufacturer details (name and phone number)

STEP 17 - DISPLAY the details of the available laptop stock

STEP 18- INPUT the id of the laptop to purchase

STEP 19 - IF id of the laptop is less than or equals 0 or greater than 5, go to STEP 18, else go to STEP 20

STEP 20- INPUT the quantity of laptops to be purchased

Step 21- IF the quantity of the laptop is zero, go to step 20, ELSE go to step 22

Step 22- DISPLAY message to ask if the user wants to continue purchasing (y/n)

STEP 23- IF user input is "y", go to STEP 18, else go to STEP 24

STEP 24 - Generate the purchase receipt and go to step 2

STEP 25- IF user input is 3, go to step 26, else if go to step 4, else go to STEP 15

STEP 26- END the program

3. Flowchart

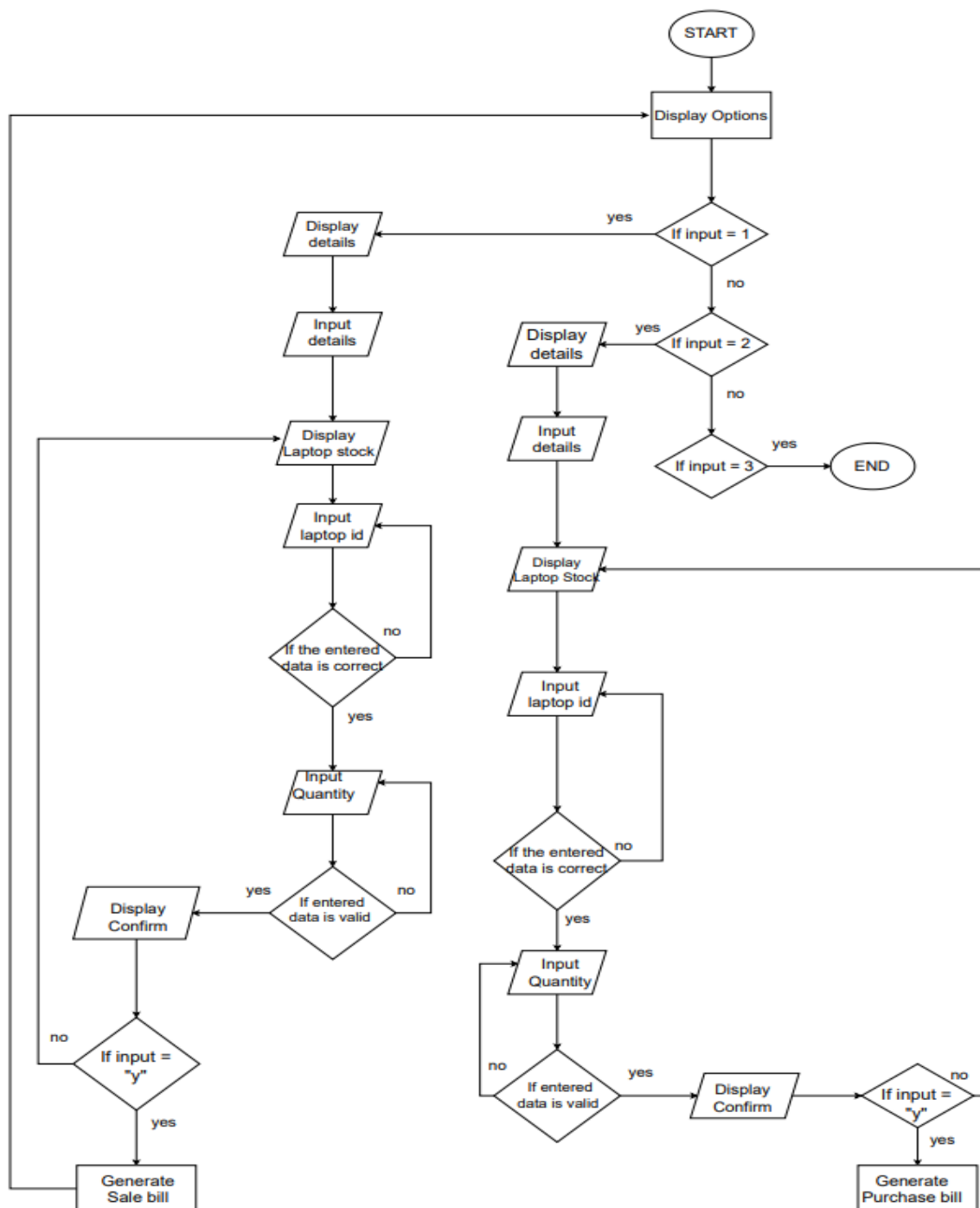


Figure 1: Flowchart of the program.

4. Pseudocode

4.1 For main

IMPORT datetime

IMPORT operations

IMPORT read

IMPORT write

```
PRINT "\n\n\n"
```

```
PRINT "\t\t\t\t\t\t\t!!!!!!LAPTOP INDUSTRIES !!!!!!"
```

PRINT “\n”

CALL stored_in_dictionary() function from read file

CALL display_stock() function to display the available laptop stock from read file

ASSIGN laptop_data to read laptop data from "laptop_data.txt" file

SET loop variable to True

WHILE loop is True

PRINT “\n”

PRINT "Press 1 to sale laptop."

PRINT "Press 2 to purchase laptop."

```
PRINT "Press 3 to Exit."
```

PRINT “/n”

SET loop_check variable to True

WHILE loop_check is True

TRY

CONVERT input ("Enter the option to continue: ") to an integer of the variable

user_input

SET loop_check to False

EXCEPT

PRINT "Please enter a valid detail!"

PRINT "\n"

IF user_input is 1

PRINT "To generate a bill, provide the details:"

PRINT "\n"

INPUT name as "Enter name of the customer: "

PRINT "\n"

INPUT phone as "Enter the phone number: "

PRINT "\n"

CREATE user_purchased_laptops list

SET loop_1 variable to True

WHILE loop_1 is True

CALL display_stock() function from read file

CALL the stored_in_dictionary function to read the laptop data from file

CALL the validate_id function to get the user's selected laptop ID and store it in
the user_id variable

IF the quality of the selected laptop is zero

PRINT "The product is currently out of stock!!!"

SET loop_1 to True

ELSE

ASSIGN quantity_details to laptop_data[user_id][3]

CALL the sales_quantity function in the quantity variable

PRINT "\n"

UPDATE laptop_data[user_id][3] by subtracting quantity

CALL the updating_function to save the updated laptop data to the file

ASSIGN product_name to laptop_data[user_id][0]

ASSIGN product_brand to laptop_data[user_id][1]

ASSIGN selected_laptop_quantity to quantity

ASSIGN unit_price to laptop_data[user_id][2]

ASSIGN selected_item_price to unit_price without "\$" sign

ASSIGN total_price to the product of selected_item_price and selected_laptop_quantity

APPEND [product_name, product_brand, selected_laptop_quantity, unit_price, total_price] to user_purchased_laptops list

INPUT buy_more as "do you want to continue (y/n): " converted to uppercase

IF buy_more == "Y"

SET loop_1 to True

ELSE

CALL sales_receipt(user_purchased_laptops, name, phone)

function to generate the sales receipt

SET loop_1 to False

ELIF user_input is 2

PRINT "To generate a bill, provide the details:"

PRINT "\n"

INPUT name as "Enter name of the manufacturer: "

PRINT "\n"

INPUT phone as "Enter the phone number: "

PRINT "\n"

CREATE user_purchased_laptops list

SET loop_1 variable to True

WHILE loop1 is True

CALL display_stock() function from read file

ASSIGN laptop_data to the stored_in _dictionary

ASSIGN user_id to the validated_id

ASSIGN quantity_details to laptop_data[user_id][3]

CALL purchase_quantity(quantity_details) function to get purchase quantity from the user

PRINT "\n"

UPDATE laptop_data[user_id][3] by adding quantity

CALL updating_function(laptop_data) function to update the data

ASSIGN product_name to laptop_data[user_id][0]

ASSIGN product_brand to laptop_data[user_id][1]

ASSIGN selected_laptop_quantity to quantity

ASSIGN unit_price to laptop_data[user_id][2]

ASSIGN selected_item_price to unit_price without "\$" sign

ASSIGN total_price to the product of selected_item_price and

selected_laptop_quantity

APPEND [product_name, product_brand, selected_laptop_quantity, unit_price, total_price] to user_purchased_laptops list

INPUT buy_more as "do you want to continue (y/n): " converted to uppercase

IF buy_more == "Y"

SET loop_1 to True

ELSE

CALL purchase_receipt(user_purchased_laptops, name, phone)

 function to

 generate the purchase receipt

SET loop_1 to False

ELIF user_input == 3

SET loop to False

PRINT "Thank you for your time with us!!!"

PRINT "\n"

ELIF

PRINT "\n"

PRINT "Invalid choice"

PRINT "Options ID are: 1, 2, and 3"

END WHILE

4.2 For operations file

DEFINE the function validate_id with parameter laptop_data

SET loop variable to True

WHILE loop is True

TRY

PRINT "\n"

CONVERT user_id in integer and input "Please provide id of the laptop: "

PRINT \n

SET loop to False

EXCEPT

PRINT "Please enter a valid detail."

WHILE user_id <=0 or user_id > len(laptop_data):

PRINT \n

PRINT "Invalid choice! Product ID are: 1,2 and 3"

SET loop to True

WHILE loop is True

TRY

PRINT \n

CONVERT user_id in integer and input "Please provide id of the
laptop: "

```
        PRINT \n

        SET loop to False

    EXCEPT

        PRINT "Please enter a valid detail.

RETURN user_id

END FUNCTION

DEFINE function sales_quantity with the parameter quantity_details
    SET loop to True
    WHILE loop is True
        TRY
            PRINT "\n"
            CONVERT input("Please provide quantity of laptop you want to sell: ") to an
            integer of the variable quantity
            PRINT "\n"
            SET loop to False
        EXCEPT
            PRINT "Please enter a valid detail."

    WHILE quantity <= 0 OR quantity > CONVERT quantity_details to an integer
        PRINT "\n"
        PRINT "Invalid choice!"
        SET loop to True
        WHILE loop is True
            TRY
                PRINT "\n"
                CONVERT input("Please provide quantity of laptop you want to sell: ") to an
                integer of the variable quantity
```



```
    PRINT "\n"  
    SET loop to False  
EXCEPT  
    PRINT "Please enter a valid detail."
```

```
RETURN quantity  
END FUNCTION
```

DEFINE function purchase_quantity with parameter quantity_details

```
    SET loop to True  
    WHILE loop is True  
        TRY  
            PRINT "\n"  
            CONVERT input("Please provide quantity of laptop you want to purchase: ") to  
            an integer of the variable quantity  
            PRINT "\n"  
            SET loop to False  
        EXCEPT  
            PRINT "Please enter a valid detail."
```

```
WHILE quantity <= 0  
    PRINT "\n"  
    PRINT "Invalid choice!"  
    SET loop to True  
    WHILE loop is True  
        TRY  
            PRINT "\n"  
            CONVERT input("Please provide quantity of laptop you want to purchase: ")  
            to an integer of the variable quantity  
            PRINT "\n"  
            SET loop to False
```

EXCEPT

PRINT "Please enter a valid detail."

RETURN quantity

END FUNCTION

4.3 For Read file

DEFINE the function stored_in_dictionary()

OPEN the "laptop_data.txt" file in read mode and store it in the file variable

CREATE an empty dictionary named l_data to store the laptop data

ASSIGN a variable i_d to 1

FOR lines in file

ASSIGN the lines variable from the lines.replace

UPDATE the l_data with {i_d: lines.split(",")}

ASSIGN i_d variable as 1

END FOR

CLOSE the file

RETURN the l_data dictionary

DEFINE the function display_stock():

PRINT "-----
-----"

PRINT "S.N. \t\t LAPTOP NAME CompanyName\t\tPrice\t\tQuantity\t\tProcessor
\t\tGraphics Card"

```
PRINT "-----  
-----"
```

OPEN the "laptop_data.txt" file in read mode and store it in the file variable.

ASSIGN a variable i_d to 1

FOR lines in file

```
PRINT i_d, "\t\t" + lines.replace(",","\t\t")
```

ASSIGN i_d variable by i_d+1

END FOR

```
PRINT "-----  
"
```

CLOSE the file

```
PRINT "\n"
```

4.4 For Write File

DEFINE the function updating_function(laptop_data):

OPEN the "laptop_data.txt" file in write mode and store it in the file variable

FOR each dictionary value (laptop data) in the laptop_data dictionary

CONVERT each element to a string and **CONCATENATES** them with
commas to create a formatted line

WRITE a newline character

END FOR

CLOSE the file

DEFINE the function purchase_receipt with parameter user_purchased_laptops, name,
phone

INITIALIZE total variable with the value 0

INITIALIZE VAT with the value 0.3

FOR each item i in the user_purchased_laptops:

ASSIGN total price to an integer and extract the fifth element (index 4) of the
item i and add it to the current value of total

END FOR

SET VAT_AMT to the product of total and VAT

SET grand_total to the sum of total and VAT_AMT

PRINT "\n"

PRINT "\t \t \t \t laptop shop bill"

```

PRINT "\n"

PRINT "\t\t Kamalpokhari, Kathmandu | Phone No: 9849408157"

PRINT "\n"

PRINT "-----"

PRINT "laptop details are:"

PRINT "-----"

PRINT "Name of manufacturer:" + str(name)

PRINT "contact number:" + str(phone)

PRINT "-----"

PRINT "\n"

PRINT "Purchase Details are:"

PRINT "-----"

PRINT "Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t\t Total"

PRINT "-----"

FOR each item i in the user_purchased_laptops:

    PRINT "i[0],"\t\t\t", i[1],"\t\t\t",i[2],"\t\t\t", "$", i[3],"\t\t\t", "$", i[4]"

END FOR

PRINT "-----"

PRINT "Your VAT cost is $", VAT_AMT

PRINT "Grand Total: $" + str(grand_total)

OPEN a new file with str(name) + ".txt" in write mode

```

WRITE "\n" into the file

WRITE "\t \t \t \t \t laptop shop bill \n" into the file

WRITE "\n" into the file

WRITE "\t \t kamalpokhari, kathmandu | Phone No: 9849408157 \n" into the file

WRITE "\n" into the file

WRITE "-----" into the file

WRITE "laptop details are: \n"

WRITE "-----" into the file

WRITE "Name of Customer:" + str(name) + "\n" into the file

WRITE "Contact number: " + str(phone) + "\n" into the file

WRITE "-----" into the file

WRITE "\n" into the file

WRITE "Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t\t Total" into the

file

WRITE "-----" into the file

FOR each item i in the user_purchased_laptops:

WRITE (str(i[0]) + "\t\t" + str(i[1]) + "\t\t" + str(i[2]) + "\t\t" + "\$" + str(i[3]) +

"\$" + str(i[4]) + "\n") into the file

END FOR

WRITE "-----" into the file

WRITE "\n" into the file

WRITE "Your VAT is:" + "" +str(VAT_AMT)+"\n" into the file

WRITE "\n" into the file

WRITE "Grand Total : \$" +str(grand_total)+"\n" into the file

WRITE "\n" into the file

CLOSE the file

DEFINE sales_receipt with parameter user_purchased_laptops,name,phone

ASSIGN variable total as 0

ASSIGN variable shipping_cost as 500

FOR each item i in the user_purchased_laptops:

ASSIGN total price to an integer and extract the fifth element (index 4) of
 the item i and add it to the current value of total

END FOR

SET grand_total to the sum of total and shipping_cost

PRINT "\n"

PRINT "\t \t \t bill"

PRINT "\n"

PRINT "\t \t Kamalpokhari, Kathmandu | Phone No: 9849408157"

PRINT "\n"

PRINT "-----"

PRINT "laptop details are:"


```
PRINT "-----"

PRINT "Name of customer:" + str(name)

PRINT "Contact number:" +str(phone)

PRINT "-----"

PRINT "\n"

PRINT "Purchase Details are:"

PRINT "-----"

PRINT "Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t Total"

PRINT "-----"

FOR each item i in the user_purchased_laptops:

    PRINT i[0], "\t\t", i[1], "\t", i[2], "\t", "$", i[3], "\t\t", "$", i[4]

END FOR

PRINT "-----"

PRINT "Your shipping cost is $", shipping_cost

PRINT " Grand Total: $" + str(grand_total)

OPEN a new file with str(name) + ".txt" in write mode

WRITE "\n" into the file

WRITE "\t \t \t \t \t bill \n" into the file

WRITE "\n" into the file

WRITE "\t \t Kamalpokhari, Kathmandu | Phone No: 9849408157 \n" into the file

WRITE "\n" into the file
```

```
WRITE "-----" into the file

WRITE "laptop details are: \n" into the file

WRITE "-----" into the file

WRITE "Name of Customer: " + str(name) + "\n"

WRITE "Contact number: " + str(phone) + "\n"

WRITE "-----" into the file

WRITE "\n" into the file

WRITE "Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t Total")into the file

WRITE "-----" into the file

FOR each item i in the user_purchased_laptops:

    WRITE str(i[0]) + "\t" + str(i[1]) + "\t" + str(i[2]) + "\t" + str(i[3]) + "\t" + "$" + str(i[4]) +

    "\n" into the file

END FOR

WRITE "-----" into the file

WRITE "\n" into the file

WRITE "Your shipping cost is:" + "" + str(shipping_cost) + "\n" into the file

WRITE "\n" into the file

WRITE "Grand Total : $" + str(grand_total) + "\n" into the file

WRITE "\n"

CLOSE the file
```

5. Data Structure

Data structures provide a structured and organized approach to storing and managing different types of data so that they can be easily modified and accessed in the future.

Data structures can be categorized into two main types: primitive and non-primitive.

5.1 Primitive data type

Primitive data structures are the fundamental ways of representing data and consist of simple values like integers, floating-point numbers, characters, and Booleans.

- a. **Integers** - The integer data type is denoted by the "int" keyword. The integer data type is a fundamental data type used to represent whole numbers in programming. It allows for the storage and manipulation of whole numbers, both positive and negative, without any fractional or decimal parts.

```
print( \n )
loop = False
apt:
print("Please enter a valid detail.")
quantity <=0 or quantity > int(quantity_details):
nt("\n")
nt("Invalid choice! ")
```

Figure 2: Integer data structure of the program.

- b. **String** - The string data type is a fundamental data type used to represent sequences of characters in programming. It allows for the storage and manipulation of textual data, such as words, sentences, or any combination of characters. Strings are commonly used for representing and working with textual information.

```
file.write("\n")
file.write("Your VAT is:" + "" +str(VAT_AMT)+"\n")
file.write("\n")
file.write("Grand Total : $" +str(grand_total)+"\n")
file.write("\n")
```

Figure 3: String data structure of the program.

- c. Float - A float, also known as a floating-point number, is a data type in programming that represents numeric values with fractional parts or decimal points. Floats can represent a wide range of numbers, including both integers and numbers with fractional parts. For example, 1.29 and 0.123 are both examples of float values.
- d. Boolean - Boolean is a basic data type that has the ability to store the values True and False, which can be identical to the numbers 1 and 0.

For example:

```
x = 10
y = 5
is_greater = x > y # The condition "x > y" is True since 10 is greater than 5.
is_equal = x == y # The condition "x == y" is False since 10 is not equal to 5.
print(is_greater) # Output: True
print(is_equal) # Output: False
```

5.2 Non-primitive data type

Non-primitive data structures contain a variety of values in various forms, making them more complex and elaborate. Arrays, lists, dictionaries, and sets are examples of non-primitive data structures that enable more complex data organization and manipulation.

- a. Lists - A list is an ordered collection of data that can be modified and can contain elements of the same or different types. Lists are represented by enclosing the elements within square brackets and separating them with commas. It allows for various operations, such as adding, removing, or modifying elements, without altering the fundamental identity of the list itself.

```
total_price = int(selected_item_price)*int(selected_laptop_quantity)
user_purchased_laptops.append([product_name,product_brand, selected_laptop_quantity,unit_price,total_price])
```

Figure 4: List data structure of the program.

- b. **Dictionary:** Dictionaries basically store 'key-value' pairs. The 'key' identifies an item, and the 'value' stores the item's value. The 'key-value' pairs are separated by commas, and the values are separated from the keys using a colon ':' character. It allows users to add, remove, or change existing key-value pairs in a dictionary.

```
1  stored_in_dic
   file = open("
   l_data = {}
   i_d=1
```

Figure 5: Dictionary data structure of the program.

- c. **Tuples:** Tuples are used to store multiple items in a single variable. A tuple is a collection which is ordered and unchangeable. Tuples are written with round brackets.

For example:

```
fruits = ("apple", "banana", "cherry")
print(fruits)
```

- d. The `range()` function returns a sequence of numbers, starting from 0 by default, and increments by 1 (by default), and stops before a specified number.

For example:

```
x = range(5, 8)
for n in x:
    print(n)
```

6. Program

Format - The program starts by prompting for the shop name and then displays a menu of options, which includes showing a list of all laptops that are currently available along with information about them, including their company name, price, number, CPU, and graphics card. It also provides the options to sell a laptop, purchase a laptop, or exit the program.

```
!!!!!!LAPTOP INDUSTRIES !!!!!!
```

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	8	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	16	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

```
Press 1 to sale laptop.  
Press 2 to purchase laptop.  
Press 3 to Exit.  
  
Enter the option to continue:
```

Figure 6: Screenshot of the program's format.

Sale laptop - Option 1 is for selling a laptop. When the user selects option 1, they will be asked to provide details such as their name and phone number. After that, the program displays all the available laptops along with their information. The user should enter the ID of the laptop they want to sell and specify the quantity. Once the laptop is sold, the program reduces the quantity accordingly. The program then asks if the user wants to continue. If the user decides not to continue, the program generates a bill for the transaction.

```

Enter the option to continue: 1

To generate a bill, provide the details:

Enter name of the customer: Rama Shrestha

Enter the phone number: 9847630146

-----
S.N.      Laptop Name      Company Name      Price      Quantity      Processor      Graphics Card
-----
1         Razer Blade      Razer            $2000      8             i7 7th Gen     GTX 3060
2         XPS              Dell             $1976      16            i5 9th Gen     GTX 3070
3         Alienware        Alienware        $1978      15            i5 9th Gen     GTX 3070
4         Swift 7          Acer             $900       25            i5 9th Gen     GTX 3070
5         Mackbook Pro 16  Apple           $3500      20            i5 9th Gen     GTX 3070
-----

Please provide id of the laptop: 2

Please provide quantity of laptop you want to sell: 4

do you want to continue y nn

bill

Kamalpokhari, Kathmandu | Phone No: 9849408157

-----
laptop details are:
-----
Name of customer:Rama Shrestha
Contact number:9847630146
-----

Purchase Details are:
-----
Item Name      Brand      Total Quantity      Unit Price      Total
-----
XPS            Dell       4                   $1976           $ 7904
-----
Your shipping cost is $ 500
Grand Total: $8404

Press 1 to sale laptop.
Press 2 to purchase laptop.
Press 3 to Exit.

Enter the option to continue:

```

Figure 7: Screenshot of Sale laptop.

Purchase laptop - Option 2 is for purchasing a laptop. When the user selects option 2, they will be asked to provide details, such as their name and phone number. After that, the program displays all the available laptops along with their information. The user should enter the ID of the laptop they want to purchase and specify the quantity. Once the laptop is purchased, the program adds the quantity accordingly. The program then asks if the user wants to continue. If the user decides not to continue, the program generates a bill for the transaction.

Enter the option to continue: 2

To generate a bill, provide the details:

Enter name of the manufacturer: Binod Shrestha

Enter the phone number: 9745639017

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	8	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	12	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Please provide id of the laptop: 1

Please provide quantity of laptop you want to purchase: 10

do you want to continue y nn

```

                                laptop shop bill

                                Kamalpokhari, Kathmandu | Phone No: 9849408157

-----
laptop details are:
Name of manufacturer:Binod Shrestha
contact number:9745639017
-----

Purchase Details are:
-----
Item Name      Brand      Total Quantity      Unit Price      Total
-----
Razer Blade    Razer      10                   $ $2000          $ 20000
-----

Your VAT cost is $ 2600.0
Grand Total: $22600.0

Press 1 to sale laptop.
Press 2 to purchase laptop.
Press 3 to Exit.

Enter the option to continue:

```

Figure 8: Screenshot of purchase laptop.

Exit - Option 3 is for exiting the program. When the user selects option 3, a 'Thank you' message appears.

```

Press 1 to sale laptop.
Press 2 to purchase laptop.
Press 3 to Exit.

Enter the option to continue: 3

Thank you for your time with us!!!

```

Figure 9: Screenshot of exiting the program.

Before purchase and sale: Displaying all the available laptops along with their quantity and other information.

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	8	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	16	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Figure 10: Screenshot of before purchase and sale

After selling 4 XPS, the program reduces the quantity.

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	8	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	12	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Figure 11: Screenshot of quantity of laptop after selling.

After purchasing 10 Razer Blade laptops, the program updates the quantity.

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	12	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Figure 12: Screenshot of quantity of laptop after purchasing.

File before purchase and sale

Name	Status	Date modified	Type	Size
__pycache__	✓	7/26/2023 8:34 PM	File folder	
laptop_data	✓	7/26/2023 9:12 PM	Text Document	1 KB
main	✓	7/26/2023 8:34 PM	Python Source File	5 KB
operations	✓	7/26/2023 7:03 PM	Python Source File	3 KB
Rama Shrestha	✓	7/26/2023 9:11 PM	Text Document	1 KB
read	✓	7/26/2023 5:24 PM	Python Source File	2 KB
write	✓	7/26/2023 8:34 PM	Python Source File	8 KB

Figure 13: Screenshot of the file before purchase and sale.

After purchase and sale

Name	Status	Date modified	Type	Size
__pycache__	✓	7/26/2023 8:34 PM	File folder	
Binod Shrestha	✓	7/26/2023 9:12 PM	Text Document	1 KB
laptop_data	✓	7/26/2023 9:12 PM	Text Document	1 KB
main	✓	7/26/2023 8:34 PM	Python Source File	5 KB
operations	✓	7/26/2023 7:03 PM	Python Source File	3 KB
Rama Shrestha	✓	7/26/2023 9:11 PM	Text Document	1 KB
read	✓	7/26/2023 5:24 PM	Python Source File	2 KB
write	✓	7/26/2023 8:34 PM	Python Source File	8 KB

Figure 14: Screenshot of file after purchase and sale.

Bill generated in text file after purchase.

```

                                laptop shop bill

                                kamalpokhari, kathmandu | Phone No: 9849408157

-----
laptop details are:
-----
Name of Customer:Binod Shrestha
Contact number: 9745639017
-----

Item Name           Brand           Total Quantity           Unit Price           Total
-----
Razer Blade         Razer              10                        $2000                $20000
-----
Your VAT is:2600.0
Grand Total : $22600.0

```

Figure 15: Screenshot of after purchase in text file

Bill generated in text file after sale.

```

                                bill

                                Kamalpokhari, Kathmandu | Phone No: 9849408157

-----
laptop details are:
-----
Name of Customer: Rama Shrestha
Contact number: 9847630146
-----

Item Name           Brand           Total Quantity           Unit Price           Total
-----
XPS                  Dell              4                        $1976                $7904
-----
Your shipping cost is:500
Grand Total : $8404

```

Figure 16: Screenshot of after sell in text file

7. Testing

7.1 Test 1: Show implementation of try, except

Test	1
Objective	To verify the functionality of try-except block in handling invalid input and displaying the appropriate error message
Action	<ul style="list-style-type: none">- Opened the program in IDLE.- Provided invalid input.
Expected Result	The program should display the error message "Invalid choice." Options ID are 1, 2 and 3.
Actual Result	The program successfully displayed the expected error message, stating Invalid choice. Options ID are 1, 2 and 3.
Conclusion	The test was successful

Table 1: Test 1 - Verify the functionality of try- except block.

```
!!!!!!LAPTOP INDUSTRIES !!!!!!
```

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	15	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	9	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	23	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	12	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	15	i5 9th Gen	GTX 3070

```
Press 1 to sale laptop.  
Press 2 to purchase laptop.  
Press 3 to Exit.  
  
Enter the option to continue: 4  
  
Invalid choice  
Options ID are: 1,2 and 3  
  
Press 1 to sale laptop.  
Press 2 to purchase laptop.  
Press 3 to Exit.  
  
Enter the option to continue: |
```

Figure 17: Screenshot from Test 1

7.2 Test 2: Selection purchase and sale of laptops

Test	2
Objective	To verify the program's handling of negative values and non-existent value during the selection of laptop purchase and sale.
Action	<ul style="list-style-type: none"> - Opened the program in IDLE. - Selected the laptop name as "Alienware". - Provided invalid input.
Expected Result	<p>When a negative value is chosen for purchase, the program should display the message "Please enter the valid detail."</p> <p>When non-existent value is chosen for purchase, the program should display the message "Invalid choice! Product ID are: 1,2,3,4 and 5."</p>
Actual Result	The program successfully displayed the expected message when a negative value and non-existent value was selected for purchase.
Conclusion	The test was successful.

Table 2: Test 2 - Verify the program's handling of negative value and non-existing value as input.

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	15	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	9	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	23	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	12	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	15	i5 9th Gen	GTX 3070

Please provide id of the laptop: Alienware
Please enter a valid detail.

Please provide id of the laptop:

Figure 18: Screenshot from Test 2 - Negative value

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	15	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	9	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	23	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	12	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	15	i5 9th Gen	GTX 3070

Please provide id of the laptop: 6

Invalid choice! Product ID are: 1,2,3,4 and 5

Please provide id of laptop: |

Figure 19: Screenshot from Test 2 - Non-existing value

7.3 Test 3: File generation for laptop purchase

Test	3
Objective	To verify the functionality of generating a file for the purchase of laptops.
Action	<ul style="list-style-type: none"> - Opened the program in IDLE. - Provide the id of the laptop as 2 and set the quantity to 10. - Provide the id of the laptop as 3 and set the quantity to 5. - Provide the id of the laptop as 5 and set the quantity to 2.
Expected Result	The file should be generated for multiple purchased laptops. The program should generate a file presenting the sale information, including the item, brand, total price, VAT cost and grand total, in an organized manner.
Actual Result	The file was generated for multiple purchased laptop. The program successfully generated a file presenting the sale information, including the item, brand, total price, VAT cost and grand total, in an organized manner.
Conclusion	The test was successful.

Table 3: Test 3 - Verification of laptop purchase file generation.

!!!!!!LAPTOP INDUSTRIES !!!!!

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	12	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Press 1 to sale laptop.
Press 2 to purchase laptop.
Press 3 to Exit.

Enter the option to continue: 2

To generate a bill, provide the details:

Enter name of the manufacturer: Anmol

Enter the phone number: 985342886

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	12	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Please provide id of the laptop: 2

Please provide quantity of laptop you want to purchase: 10

do you want to continue y ny

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	22	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	15	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Please provide id of the laptop: 3

Please provide quantity of laptop you want to purchase: 5

do you want to continue y ny

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	22	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	20	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	20	i5 9th Gen	GTX 3070

Please provide id of the laptop: 5

Please provide quantity of laptop you want to purchase: 2

do you want to continue y nn

laptop shop bill

Kamalpokhari, Kathmandu | Phone No: 9849408157

laptop details are:

Name of manufacturer:Anmol
contact number:985342886

Purchase Details are:

Item Name	Brand	Total Quantity	Unit Price	Total
XPS	Dell	10	\$ \$1976	\$ 19760
Alienware	Alienware	5	\$ \$1978	\$ 9890
Mackbook Pro 16	Apple	2	\$ \$3500	\$ 7000

Your VAT cost is \$ 4764.5
Grand Total: \$41414.5

Press 1 to sale laptop.
Press 2 to purchase laptop.
Press 3 to Exit.

Enter the option to continue:

Figure 20: Screenshot from Test 3 - Showing the laptop's complete purchase process and output.

__pycache__	✓	7/26/2023 8:34 PM	File folder	
Anmol	✓	7/27/2023 9:19 AM	Text Document	2 KB
laptop_data	✓	7/27/2023 9:19 AM	Text Document	1 KB
main	✓	7/26/2023 8:34 PM	Python Source File	5 KB
operations	✓	7/26/2023 7:03 PM	Python Source File	3 KB
read	✓	7/26/2023 5:24 PM	Python Source File	2 KB
write	✓	7/26/2023 8:34 PM	Python Source File	8 KB

laptop shop bill				
kamalpokhari, kathmandu Phone No: 9849408157				

laptop details are:				

Name of Customer:Anmol				
Contact number: 985342886				

Item Name	Brand	Total Quantity	Unit Price	Total

XPS	Dell	10	\$1976	\$19760
Alienware	Alienware	5	\$1978	\$9890
Mackbook Pro 16	Apple	2	\$3500	\$7000

Your VAT is:4764.5				
Grand Total : \$41414.5				

Figure 21: Screenshot from Test 3 - Information about purchased laptops in a text file.

7.4 Test 4: File generation for laptops sales process

Test	4
Objective	To verify the functionality of generating a file for the sale of laptops.
Action	<ul style="list-style-type: none"> - Opened the program in IDLE. - Provide the id of the laptop as 4 and set the quantity to 2. - Provide the id of the laptop as 5 and set the quantity to 1. - Provide the id of the laptop as 1 and set the quantity to 3.
Expected Result	The file should be generated for multiple sale laptops. The program should generated a file presenting the sale information, including the item, brand, total price, shipping cost and grand total, in an organized manner.
Actual Result	The file should be generated for multiple sale laptops. The program successfully generated a file presenting the sale information, including the item, brand, total price, shipping cost and grand total, in an organized manner.
Conclusion	The test was successful.

Table 4: Test 4 - Verification of laptop sale file generation.

Press 1 to sale laptop.
 Press 2 to purchase laptop.
 Press 3 to Exit.

Enter the option to continue: 1

To generate a bill, provide the details:

Enter name of the customer: Anish

Enter the phone number: 9864536983

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	22	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	20	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	25	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	22	i5 9th Gen	GTX 3070

Please provide id of the laptop: 4

Please provide quantity of laptop you want to sell: 2

do you want to continue y ny

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	22	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	20	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	23	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	22	i5 9th Gen	GTX 3070

Please provide id of the laptop: 5

Please provide quantity of laptop you want to sell: 1

do you want to continue y ny

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	18	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	22	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	20	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	23	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	21	i5 9th Gen	GTX 3070

Please provide id of the laptop: 1

Please provide quantity of laptop you want to sell: 3

do you want to continue y nn

bill

Kamalpokhari, Kathmandu | Phone No: 9849408157

laptop details are:

Name of customer:Anish
Contact number:9864536983

Purchase Details are:

Item Name	Brand	Total Quantity	Unit Price	Total
Swift 7	Acer	2	\$900	\$ 1800
Mackbook Pro 16	Apple	1	\$3500	\$ 3500
Razer Blade	Razer	3	\$2000	\$ 6000

Your shipping cost is \$ 500
Grand Total: \$11800

Figure 22: Screenshot from Test 4 - Showing the laptop's complete sale process and output.

Name	Status	Date modified	Type	Size
__pycache__	✓	7/26/2023 8:34 PM	File folder	
Anish	✓	7/27/2023 10:14 AM	Text Document	2 KB
laptop_data	✓	7/27/2023 10:14 AM	Text Document	1 KB
main	✓	7/26/2023 8:34 PM	Python Source File	5 KB
operations	✓	7/26/2023 7:03 PM	Python Source File	3 KB
read	✓	7/26/2023 5:24 PM	Python Source File	2 KB
write	✓	7/26/2023 8:34 PM	Python Source File	8 KB

bill				
Kamalpokhari, Kathmandu Phone No: 9849408157				

laptop details are:				

Name of Customer: Anish				
Contact number: 9864536983				

Item Name	Brand	Total Quantity	Unit Price	Total

Swift 7	Acer	2	\$900	\$1800
Mackbook Pro 16	Apple	1	\$3500	\$3500
Razer Blade	Razer	3	\$2000	\$6000

Your shipping cost is:500				
Grand Total : \$11800				

Figure 23: Screenshot of Test 4 - Information about sold laptops in a text file.

7.5 Test 5: Show the update in stock of laptops.

Test	5
Objective	To verify the functionality of updating the stock of laptops.
Action	<ul style="list-style-type: none"> - Opened the program in IDLE. - Executed this action after the process of selling and adding laptops.
Result	The number of laptops in stock should reflect the updates made during the selling and adding processes.
Actual Result	The number of laptops in stock is successfully updated, reflecting the changes made during the selling and adding processes.
Conclusion	The test was successful.

Table 4: Test 5- Verifying the update in stock of laptops.

Press 1 to sale laptop.
 Press 2 to purchase laptop.
 Press 3 to Exit.

Enter the option to continue: 2

To generate a bill, provide the details:

Enter name of the manufacturer: Akash

Enter the phone number: 9865357986

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	15	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	22	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	18	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	23	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	21	i5 9th Gen	GTX 3070

Please provide id of the laptop: 2

Please provide quantity of laptop you want to purchase: 4

do you want to continue y nn

laptop shop bill

Kamalpokhari, Kathmandu | Phone No: 9849408157

laptop details are:

Name of manufacturer: Akash
 contact number: 9865357986

Purchase Details are:

Item Name	Brand	Total Quantity	Unit Price	Total
XPS	Dell	4	\$ \$1976	\$ 7904

Your VAT cost is \$ 1027.52
 Grand Total: \$8931.52

Press 1 to sale laptop.
 Press 2 to purchase laptop.
 Press 3 to Exit.

Enter the option to continue:

Figure 24: Screenshot from Test 5 - Showing the laptop's complete purchase and output.

pycache	✓	7/26/2023 8:34 PM	File folder	
Akash	✓	7/27/2023 10:47 AM	Text Document	1 KB
laptop_data	✓	7/27/2023 10:47 AM	Text Document	1 KB
main	✓	7/26/2023 8:34 PM	Python Source File	5 KB
operations	✓	7/26/2023 7:03 PM	Python Source File	3 KB
read	✓	7/26/2023 5:24 PM	Python Source File	2 KB
write	✓	7/26/2023 8:34 PM	Python Source File	8 KB

laptop shop bill				
kamalpokhari, kathmandu Phone No: 9849408157				

laptop details are:				

Name of Customer: Akash				
Contact number: 9865357986				

Item Name	Brand	Total Quantity	Unit Price	Total

XPS	Dell	4	\$1976	\$7904

Your VAT is: 1027.52				
Grand Total : \$8931.52				

Figure 25: Screenshot from Test 5 - Information about purchased laptops in a text file.

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	15	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	26	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	18	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	23	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	21	i5 9th Gen	GTX 3070

Figure 26: Screenshot from Test 5 - The amount added after purchasing the laptop.

```

!!!!!!!!LAPTOP INDUSTRIES !!!!!!!

-----
S.N.      Laptop Name    Company Name    Price    Quantity    Processor    Graphics Card
-----
1          Razer Blade    Razer           $2000     15          17 7th Gen    GTX 3060
2          XPS            Dell            $1976     22          i5 9th Gen     GTX 3070
3          Alienware      Alienware       $1978     20          i5 9th Gen     GTX 3070
4          Swift 7         Acer            $900      23          i5 9th Gen     GTX 3070
5          Mackbook Pro 16   Apple           $3500     21          i5 9th Gen     GTX 3070
-----

Press 1 to sale laptop.
Press 2 to purchase laptop.
Press 3 to Exit.

Enter the option to continue: 1

To generate a bill, provide the details:

Enter name of the customer: Binod

Enter the phone number: 9854327668

-----
S.N.      Laptop Name    Company Name    Price    Quantity    Processor    Graphics Card
-----
1          Razer Blade    Razer           $2000     15          17 7th Gen    GTX 3060
2          XPS            Dell            $1976     22          i5 9th Gen     GTX 3070
3          Alienware      Alienware       $1978     20          i5 9th Gen     GTX 3070
4          Swift 7         Acer            $900      23          i5 9th Gen     GTX 3070
5          Mackbook Pro 16   Apple           $3500     21          i5 9th Gen     GTX 3070
-----

Please provide id of the laptop: 3

Please provide quantity of laptop you want to sell: 2

do you want to continue y nn

      bill

      Kamalpokhari, Kathmandu | Phone No: 9849408157

-----
laptop details are:
-----
Name of customer:Binod
Contact number:9854327668
-----

Purchase Details are:
-----
Item Name      Brand      Total Quantity    Unit Price    Total
-----
Alienware      Alienware      2                $1978        $ 3956
-----
Your shipping cost is $ 500
Grand Total: $4456

Press 1 to sale laptop.
Press 2 to purchase laptop.
Press 3 to Exit.

Enter the option to continue:

```

Figure 27: Screenshot from Test 5 – showing the laptop's complete sale process and output.

pycache	✓	7/26/2023 8:34 PM	File folder	
Binod	✓	7/27/2023 10:37 AM	Text Document	1 KB
laptop_data	✓	7/27/2023 10:36 AM	Text Document	1 KB
main	✓	7/26/2023 8:34 PM	Python Source File	5 KB
operations	✓	7/26/2023 7:03 PM	Python Source File	3 KB
read	✓	7/26/2023 5:24 PM	Python Source File	2 KB
write	✓	7/26/2023 8:34 PM	Python Source File	8 KB

bill				
Kamalpokhari, Kathmandu Phone No: 9849408157				

laptop details are:				

Name of Customer: Binod				
Contact number: 9854327668				

Item Name	Brand	Total Quantity	Unit Price	Total

Alienware	Alienware	2	\$1978	\$3956

Your shipping cost is:500				
Grand Total : \$4456				

Figure 28: Screenshot from Test 5 - Information about sold laptops in a text file.

S.N.	Laptop Name	Company Name	Price	Quantity	Processor	Graphics Card
1	Razer Blade	Razer	\$2000	15	i7 7th Gen	GTX 3060
2	XPS	Dell	\$1976	22	i5 9th Gen	GTX 3070
3	Alienware	Alienware	\$1978	18	i5 9th Gen	GTX 3070
4	Swift 7	Acer	\$900	23	i5 9th Gen	GTX 3070
5	Mackbook Pro 16	Apple	\$3500	21	i5 9th Gen	GTX 3070

Figure 29: Screenshot from Test 5 - The amount deducted after selling the laptop.

8. Conclusion

In conclusion, the system created for the laptop store successfully handles the difficulties related to processing orders, managing inventory, and creating transaction records. The store can effectively keep track of available laptops, improve sales to customers, and place orders with manufacturers by using a program that reads and updates a text file containing laptop information.

The algorithm of the system, which is presented in steps, pseudocode, and flowcharts, offers a clear and organized method for managing numerous transactions. In order to ensure data integrity and user-friendly interaction, the application performs input validation, presenting error messages when invalid data is entered.

To verify the program's functionality, complete testing was done. The report's Testing part includes screenshots and other proof of the program's effective execution and exact generation of invoices for each transaction.

Overall, the system created provides a dependable and effective solution for the operations of the laptop store. The shop may efficiently manage its inventory, deal with orders from suppliers and consumers, produce complete transaction records, and guarantee correct stock updates by implementing this system. These features help to promote operational effectiveness, customer service quality, and customer satisfaction.

This program's development was both challenging and rewarding. I have been able to use my understanding of programming to solve a helpful, real-world challenge. Not only has this experience expanded my knowledge in these areas, but it has also given me invaluable practical experience in software development. Overall, this project has been a great learning experience that has improved my programming abilities.

9. Appendix

9.1 For Main

```
#Import necessary modules

from datetime import datetime

from operations import *

from read import *

from write import *


# display first heading of welcome message to user

print("\n\n\n")

print("\t\t\t\t\t\t\t\t\t\t!!!!!!LAPTOP INDUSTRIES !!!!!!")

print("\n")


stored_in_dictionary()

laptop_data = stored_in_dictionary()


display_stock()


#while loop for continuous functionality

loop = True
```

```
while loop == True:

    #ask user to enter correct option

    print("\n")

    print("Press 1 to sale laptop.")
    print("Press 2 to purchase laptop.")
    print("Press 3 to Exit.")

    print("\n")

    loop_check = True

    while loop_check == True:

        try:

            user_input = int(input("Enter the option to continue: "))

            loop_check = False

        except:

            print("Please enter a valid detail!")

    print("\n")

    if user_input==1:

        print("To generate a bill, provide the details: ")

        print("\n")

        name = input("Enter name of the customer: ")

        print("\n")

        phone = input("Enter the phone number: ")

        print("\n")
```

```
user_purchased_laptops = []#[name,brand,price,quantity,total  
price],[name,brand,price,quantity,total price],
```

```
loop_1 = True
```

```
while loop_1 == True:
```

```
    display_stock()
```

```
    #validateid
```

```
    #also check if quantity is zero or not
```

```
    laptop_data =stored_in_dictionary()
```

```
    user_id = validate_id(laptop_data)
```

```
    if int(laptop_data[user_id][3]) == 0:
```

```
        print("The product is currently out of stock!!!")
```

```
        loop_1 = True
```

```
    else:
```

```
        quantity_details = laptop_data[user_id][3]
```

```
        quantity = sales_quantity(quantity_details)
```

```
        print("\n")
```

```
    #update stock
```

```
    laptop_data[user_id][3] = int(laptop_data[user_id][3]) - int(quantity)
```

```
    updating_function(laptop_data)
```

```
    #pricing
```

```
    product_name = laptop_data[user_id][0]
```



```
product_brand = laptop_data[user_id][1]
selected_laptop_quantity = quantity
unit_price = laptop_data[user_id][2]
selected_item_price = laptop_data[user_id][2].replace("$", "")
total_price = int(selected_item_price)*int(selected_laptop_quantity)

user_purchased_laptops.append([product_name,product_brand,
selected_laptop_quantity,unit_price,total_price])
```

```
buy_more = input("do you want to continue y n").upper()
if buy_more == "Y":
    loop_1 = True
else:
    sales_receipt(user_purchased_laptops,name,phone)
    loop_1 = False
```

```
elif user_input==2:
    print("To generate a bill, provide the details: ")
    print("\n")
    name = input("Enter name of the manufacturer: ")
    print("\n")
    phone = input("Enter the phone number: ")
    print("\n")

    user_purchased_laptops = []
```

```
loop_1 = True
while loop_1 == True:
    display_stock()

    #validateid

    #also check if quantity is zero or not

    laptop_data = stored_in_dictionary()
    user_id = validate_id(laptop_data)

    quantity_details = laptop_data[user_id][3]
    quantity = purchase_quantity(quantity_details)
    print("\n")

    #update stock
    laptop_data[user_id][3] = int(laptop_data[user_id][3]) + int(quantity)
    updating_function(laptop_data)

    #pricing
    product_name = laptop_data[user_id][0]
    product_brand = laptop_data[user_id][1]
    selected_laptop_quantity = quantity
    unit_price = laptop_data[user_id][2]
    selected_item_price = laptop_data[user_id][2].replace("$", "")
    total_price = int(selected_item_price)*int(selected_laptop_quantity)
```

```
user_purchased_laptops.append([product_name,product_brand,  
selected_laptop_quantity,unit_price,total_price])
```

```
buy_more = input("do you want to continue y n").upper()
```

```
if buy_more == "Y":
```

```
    loop_1 = True
```

```
else:
```

```
    purchase_receipt(user_purchased_laptops,name,phone)
```

```
    loop_1 = False
```

```
elif user_input == 3:
```

```
    loop = False
```

```
    print("Thank you for your time with us!!!")
```

```
    print("\n")
```

```
else:
```

```
    print("\n")
```

```
    print("Invalid choice")
```

```
    print("Options ID are: 1,2 and 3")
```

9.2 For operations

```
def validate_id(laptop_data):
```

```
    loop = True
```

```
    while loop == True:
```

```
        try:
```

```
            print("\n")
```

```
            user_id = int(input("Please provide id of the laptop: "))
```

```
            print("\n")
```

```
            loop = False
```

```
        except:
```

```
            print("Please enter a valid detail.")
```

```
while user_id <=0 or user_id > len(laptop_data):
```

```
    print("\n")
```

```
    print("Invalid choice! Product ID are: 1,2,3,4 and 5")
```

```
    loop = True
```

```
    while loop == True:
```

```
        try:
```

```
            print("\n")
```

```
            user_id = int(input("Please provide id of laptop: "))
```

```
            print("\n")
```

```
            loop = False
```

```
        except:
```

```
            print("Please enter a valid detail.")
```

```
return user_id
```

```
def sales_quantity(quantity_details):  
    loop = True  
    while loop == True:  
        try:  
            print("\n")  
            quantity = int(input("Please provide quantity of laptop you want to sell: "))  
            print("\n")  
            loop = False  
        except:  
            print("Please enter a valid detail.")  
    while quantity <=0 or quantity > int(quantity_details):  
        print("\n")  
        print("Invalid choice! ")  
        loop = True  
        while loop == True:  
            try:  
                print("\n")  
                quantity = int(input("Please provide quantity of laptop you want to sell: "))  
                print("\n")  
                loop = False  
            except:  
                print("Please enter a valid detail.")  
    return quantity  
  
def purchase_quantity(quantity_details):
```

```
loop = True
while loop == True:
    try:
        print("\n")
        quantity = int(input("Please provide quantity of laptop you want to purchase: "))
        print("\n")
        loop = False
    except:
        print("Please enter a valid detail.")
while quantity <=0:
    print("\n")
    print("Invalid choice!")
    loop = True
while loop == True:
    try:
        print("\n")
        quantity = int(input("Please provide quantity of laptop you want to purchase:
"))
        print("\n")
        loop = False
    except:
        print("Please enter a valid detail.")
return quantity
```

9.3 For read

#function to store data in a dictionary

def stored_in_dictionary():

file = open("laptop_data.txt", "r")

l_data = {}

i_d=1

for lines in file:

lines = lines.replace("\n", "")

l_data.update({i_d:lines.split(",")})

i_d=i_d+1

file.close()

return l_data

#function to display the available stocks

def display_stock():

```
print("-----")
print("-----")
```

```
print("S.N. \t\t Laptop Name      Company Name
\t\tPrice\t\tQuantity\tProcessor\t\tGraphics Card")
```

```
print("-----")
print("-----")
```

file = open("laptop_data.txt", "r")

i_d = 1

for lines in file:

print(i_d, "\t\t" + lines.replace(", ", "\t\t"))

i_d=i_d+1

```
print("-----")
-----")
file.close()
print("\n")
```


9.4 For write

```

def updating_function(laptop_data):
    file = open("laptop_data.txt", "w")
    for values in laptop_data.values():
        file.write(str(values[0]) + "," + str(values[1]) +
", "+str(values[2])+", "+str(values[3])+", "+str(values[4])+", "+str(values[5]))
        file.write("\n")
    file.close()

def purchase_receipt(user_purchased_laptops, name, phone):
    total = 0
    VAT = 0.13

    for i in user_purchased_laptops:
        total = total + int(i[4])
    VAT_AMT = total * VAT
    grand_total = total + VAT_AMT

    print("\n")
    print("\t\t\t laptop shop bill")
    print("\n")
    print("\t\t Kamalpokhari, Kathmandu | Phone No: 9849408157")
    print("\n")
    print("-----")
    print("-----")
    print("laptop details are:")
    print("-----")
    print("-----")
    print("Name of manufacturer:" + str(name))
    print("contact number:" + str(phone))
    #datetime

```

```

    print("-----")
    print("-----")
    print("\n")
    print("Purchase Details are:")
    print("-----")
    print("-----")
    print("Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t\t Total")
    print("-----")
    print("-----")
    for i in user_purchased_laptops:

        print(i[0], "\t", i[1], "\t\t", i[2], "\t\t\t", "$", i[3], "\t\t\t", "$", i[4])
    print("-----")
    print("-----")
    print("Your VAT cost is $", VAT_AMT)
    print("Grand Total: $" + str(grand_total))

    file = open(str(name)+".txt", "w")
    file.write("\n")
    file.write("\t \t \t \t \t laptop shop bill \n")
    file.write("\n")
    file.write("\t \t kamalpokhari, kathmandu | Phone No: 9849408157 \n")
    file.write("\n")
    file.write("-----")
    print("-----\n")
    file.write("laptop details are: \n")
    file.write("-----")
    print("-----\n")
    file.write("Name of Customer:" + str(name) + "\n")
    file.write("Contact number: " + str(phone) + "\n")

```

```

        file.write("-----\n")
    -----\n")

    file.write("\n")
    file.write("Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t Total" + "\n")
    file.write("-----\n")
    -----\n")

    for i in user_purchased_laptops:

        file.write(str(i[0]) + "\t\t " + str(i[1]) + "\t\t" + str(i[2]) + "\t\t\t " + str(i[3]) + "\t\t\t" +
"$" + str(i[4]) + "\n")
        file.write("-----\n")
    -----")

    file.write("\n")
    file.write("Your VAT is:" + " " + str(VAT_AMT) + "\n")
    file.write("\n")
    file.write("Grand Total : $" + str(grand_total) + "\n")
    file.write("\n")
    file.close()

def sales_receipt(user_purchased_laptops, name, phone):
    total = 0
    shipping_cost = 500
    for i in user_purchased_laptops:

        total = total + int(i[4])
    grand_total = total + shipping_cost

    print("\n")

```

```

print("\t \t \t bill")
print("\n")
print("\t \t Kamalpokhari, Kathmandu | Phone No: 9849408157")
print("\n")
print("-----")
-----")
print("laptop details are:")
print("-----")
-----")
print("Name of customer:" + str(name))
print("Contact number:" +str(phone))

print("-----")
-----")
print("\n")
print("Purchase Details are:")
print("-----")
-----")
print("Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t Total")
print("-----")
-----")
for i in user_purchased_laptops:

    print(i[0], "\t", i[1], "\t\t", i[2], "\t\t\t", i[3], "\t\t\t", "$", i[4])
    print("-----")
    -----")

print("Your shipping cost is $", shipping_cost)
print("Grand Total: $" + str(grand_total))

file = open(str(name)+".txt", "w")
file.write("\n")

```

```
file.write("\t\t\t\t\t bill \n")
file.write("\n")
file.write("\t\t Kamalpokhari, Kathmandu | Phone No: 9849408157 \n")
file.write("\n")
file.write("-----\n")
file.write("\n")
file.write("laptop details are: \n")
file.write("-----\n")
file.write("\n")
file.write("Name of Customer: " + str(name) + "\n")
file.write("Contact number: " +str(phone) + "\n")
file.write("-----\n")
file.write("\n")
file.write("\n")
file.write("Item Name \t\t Brand \t\t Total Quantity \t\t Unit Price \t\t Total" + "\n")
file.write("-----\n")
file.write("\n")
for i in user_purchased_laptops:
    file.write(str(i[0])+"\t\t "+str(i[1])+"\t\t "+str(i[2])+"\t\t\t " +str(i[3]) + "\t\t\t " +
"$"+str(i[4])+"\n")
    file.write("-----\n")
    file.write("\n")
file.write("Your shipping cost is:" + "" +str(shipping_cost)+"\n")
file.write("\n")
file.write("Grand Total : $" +str(grand_total)+"\n")
file.write("\n")
file.close()
```