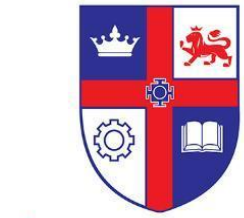
LONDON METROPOLITAN UNIVERSITY



islington college
(इस्लिङ्टन कलेज)

CS4051NI Fundamentals of Computing

60% Individual Coursework

2023 Autumn

Student Name: Rashi Maharjan

London Met ID: 22067683

College ID: NP01CP4a220113

Assignment Due Date: Friday, May 12, 2023

Assignment Submission Date: Friday, May 12, 2023

Word Count: 7375

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

Table of Contents

Table of Figures

Table of Tables

1. Introduction

This project is the development of application in Python for Laptop Shop distributor. The distributor has a stock file. It buys a product from a manufacturer and sells it to customers. So while doing buying and selling transactions, the stock gets updated by increasing and decreasing stocks respectively. The information about the available computers are managed in a text file. It is an interactive program which works in loop, displaying the available laptops and waiting for the administrator to enter details of the customers and does not close unless he administrator decides to do so. It checks the input data, displaying error messages whenever unwanted data is entered.

The aim of the project is to develop an application for a laptop rental shop to buy and sell the laptops and generate a bill along the process.

The objectives of this project is to make us able use python programming for developing an application. A customer will be able to buy and sell a laptop using this application. The program is required to read the text file, display available stock of laptops, update stocks according to transaction's nature, generate receipt of the transaction and manipulate stock in main text file on basis of transactions. This program also checks the validity of required information and exception handling.

1.1. Python

Python is the design philosophy of the general-purpose and high-level programming language that places a strong emphasis on code readability. Python's syntax enables programmers to express concepts in a simpler manner and plan designed structures to enable understandable systems on both a small and big scale. Python supports a variety of programming paradigms, including imperative, functional, and object-oriented programming. Although Python is frequently used for scripting, it also has a wide range of uses outside of scripting (Rossum, 2007).

2. Algorithm

Algorithm is the step by step process to solve the particular problem. It is a programming tool used by the programmer to develop software applications and does not have any specific rules for designing algorithm (Upadhyay, 2023).

Step 1: START

Step 2: Display the welcome message.

Step 3: Display options to purchase, sell or exit

Step 4: Input an option; 1 to purchase, 2 to sell or 3 to exit

Step 5: If 1 is taken as option, go to step 10, else to go step 6

Step 6: If 2 is taken as option, go to step 39, else go to Step 7

Step 7: If 3 is taken as input then display thank you message, else go to Step 9

Step 8: END

Step 9: If an invalid value is input, display an invalid message and go to step 3

Step 10: Create a 2D list “add_to_cart”

Step 11: Display the list of the laptops

Step 12: Input the laptop ID to be purchased

Step 13: If a valid laptop ID is chosen, go to Step 16, else go to Step 14

Step 14: Display a message that the input value is invalid

Step 15: Go to Step 11

Step 16: Input the laptop quantity to be purchased

Step 17: If quantity > 0, go to Step 20, else go to Step 18

Step 18: Display a message that the input value is invalid

Step 19: Go to Step 16

Step 20: If the laptop is in stock, go to Step 23, else go to Step 21

Step 21: Display a message that the required laptop is out of stock

Step 22: Go to Step 16

Step 23: Display the laptops list with the reduced quantity of the purchased laptop

Step 24: Append ID and quantity of the laptops purchased to the add_to_cart

Step 25: Update the stock of laptops in the dictionary

Step 26: Ask if the admin wants to purchase more laptops

Step 27: If user_input == Y, then go to Step 11, else to go Step 28

Step 28: Input the name of the user and store in the “name” variable

Step 29: Input the contact number of the user and store in the “Contact” variable

Step 30: Change the value stored in Contact variable into integer data type

Step 31: If the Contact can be transformed into integer, go to Step 33, else go to Step 29

Step 32: Display Name, Contact and Purchased date

Step 33: Display the list of the laptops purchased with the total price of the laptops

Step 34: Display Grand total price after adding VAT and shipping cost

Step 35: Write a text file with the admin’s data, laptops purchased and grand total with every transaction

Step 36: Display the thank you message

Step 37: Go to Step 3

Step 38: Create a 2D list “add_to_cart”

Step 39: Display the list of the laptops

Step 40: Input the laptop’s ID to be sold

Step 41: If a valid laptop ID is chosen, go to Step 44, else go to Step 42

Step 42: Display a message that the input value is invalid

Step 43: Go to step 39

Step 44: Input the laptop's quantity to be sold

Step 45: If quantity > 0, go to Step 48, else go to Step 46

Step 46: Display a message that the input value is invalid

Step 47: Go to step 44

Step 48: Display the updated laptop details list

Step 49: Append ID and quantity of the laptops sold to the add_to_cart

Step 50: Update the stock of laptops in the dictionary

Step 51: Ask if the admin wants to sell more laptops

Step 52: If user_input == Y, then go to Step 39, else to go Step 53

Step 53: Input the name of the user and store in the "name" variable

Step 54: Input the contact number of the user and store in the "Contact" variable

Step 55: Change the value stored in Contact variable into integer data type

Step 56: If the Contact can be transformed into integer, go to Step 57, else go to Step 54

Step 57: Display Name, Contact and Selling date

Step 58: Display the list of the laptops sold with the total price of the laptops

Step 59: Display Grand total price after adding shipping cost

Step 60: Write a text file with the admin's data, laptops sold and grand total with every transaction

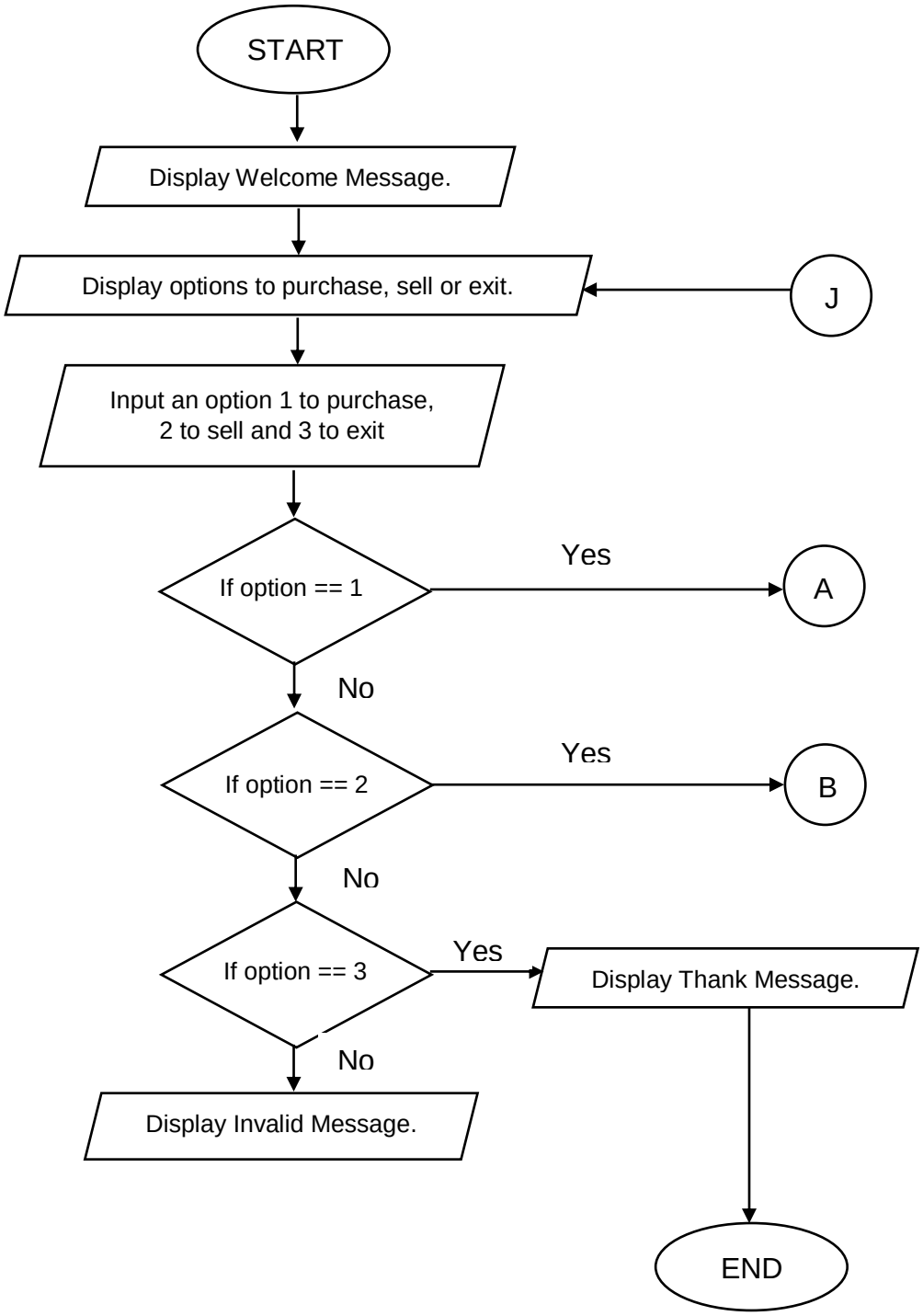
Step 61: Display the thank you message

Step 62: Go to Step 3

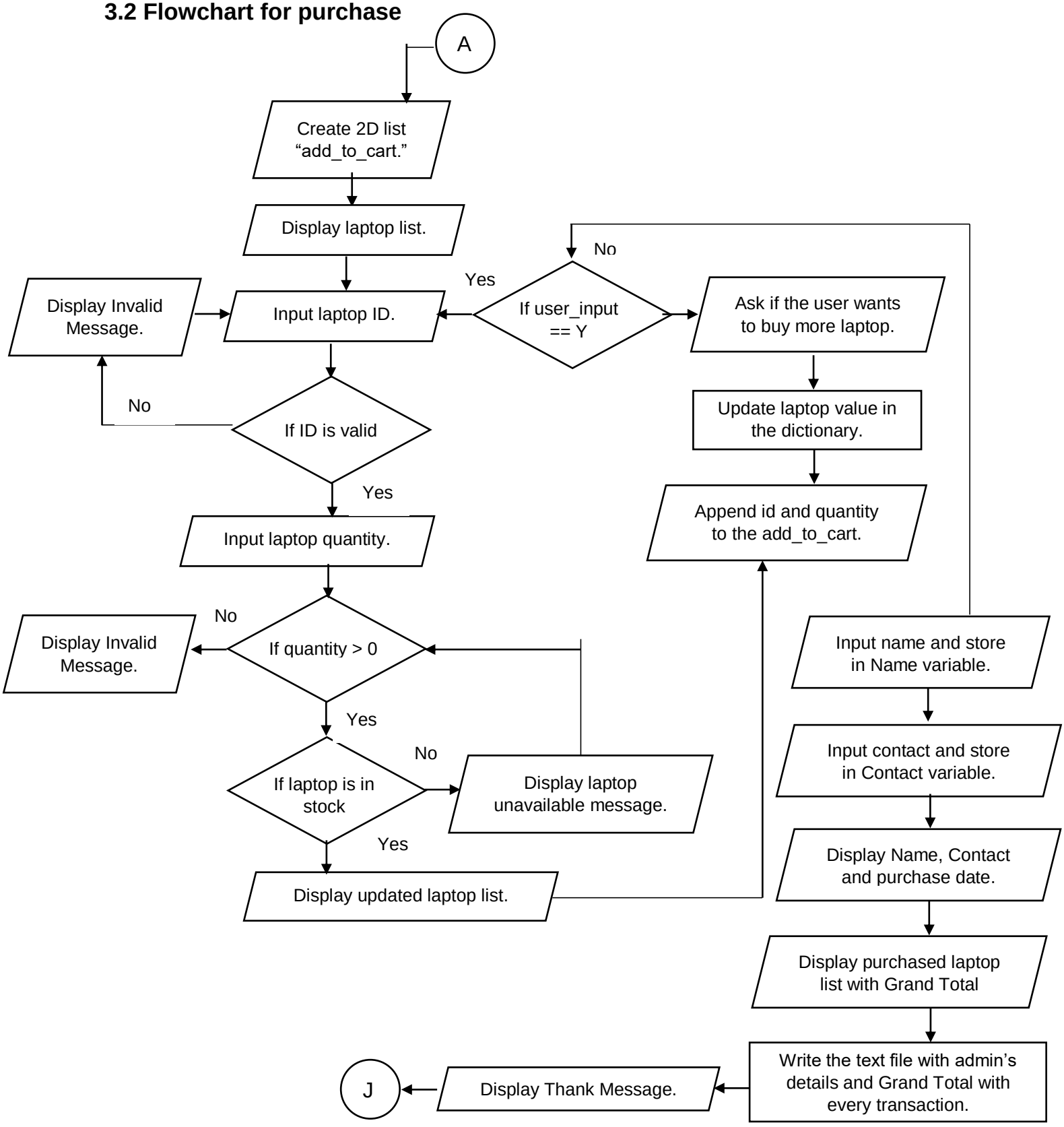
3. Flowchart

Flowchart is the pictorial representation of program. It represents the program using geometrical patterns. It is a tool and technique to find out solution of programming problems through some special symbol.

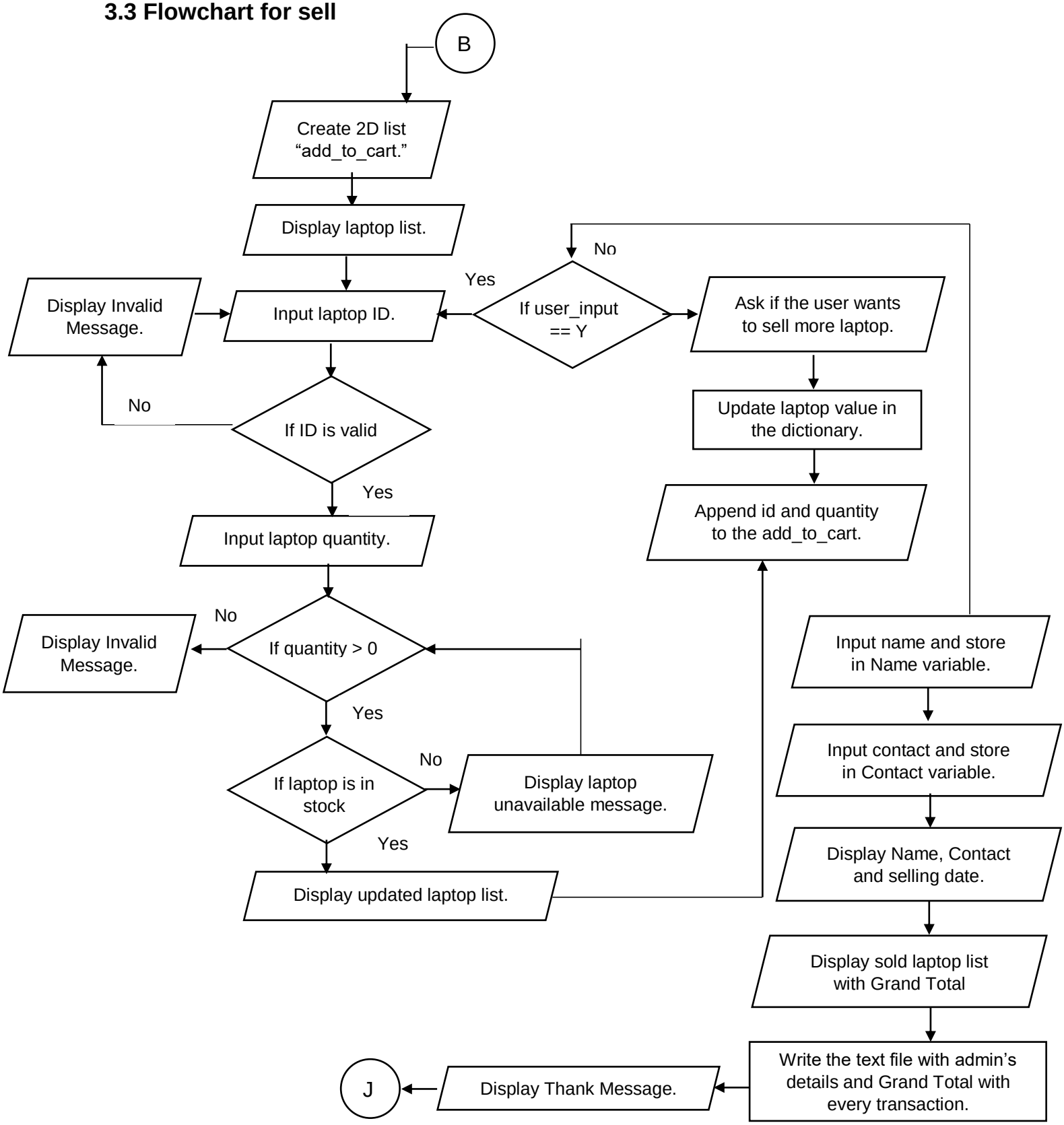
3.1 Flowchart for main



3.2 Flowchart for purchase



3.3 Flowchart for sell



4. Pseudo Code

Pseudo-code is an outline of a program that uses phrases that are typically used in spoken language and can easily be translated into actual programming statements. It is a method for describing computer programs that avoids utilizing the precise syntax and keywords of a programming language (Karatrantou & Panagiotakopoulos, 2008).

4.1 Pseudo Code for Main

```
IMPORT operation
IMPORT read

CALL welcome_message() from read
DECLARE is_running as True
WHILE is_running
    CALL option_selected() from read
    DECLARE option_selected() as False
    WHILE not option_selected
            TRY
                INITIALIZE option as int
                DECLARE option_selected as True
            EXCEPT
                CALL invalid_input() from read
                CALL option_selected() from read

    IF option is equal to 1
            CALL make_purchase() from operation
    ELIF option is equal to 2
            CALL make_sell() from operation
```

ELIF option is equal to 3

    CALL display_thanks() from read

    DECLARE is_running as False

ELSE

    CALL invalid_input() from read

4.2 Pseudo Code for Operation

```
IMPORT read
IMPORT write

DEFINE function valid_id_for_sell (value)
    DECLARE valid_data as False
    WHILE valid_data is equal to False
            TRY
                INITIALIZE ID as int
                IF ID is greater than 0 less and equal to length of value
                    IF int (value [ID] [3]) greater than 0
                        DECLARE valid_data as True
                        RETURN ID

                    ELSE
                        CALL out_of_stock() from read

                ELSE
                    CALL invalid_input() from read

            EXCEPT
                CALL invalid_input() from read

    DEFINE function valid_quantity_for_sell(value, ID)
        DECLARE quantity_validity as False
        WHILE quantity_validity is equal to False
            TRY
```

```
        INITIALIZE quantity as int
        IF quantity is less than 0 greater and equal to
        int(value[ID][3])
            DECLARE quantity_validity as True
            RETURN quantity

        ELSE
            CALL span() from read

    EXCEPT
        CALL invalid_input() from read

DEFINE function make_sell()
  DECLARE contents as read_file() from write
  DECLARE value as dictionary() with parameter as contents from write

  DECLARE add_to_cart as list
  CALL print_laptop_list() with parameter as value from write
  DECLARE continue_loop as True
  WHILE continue_loop

    DECLARE ID as valid_id_sell() with parameter as value
    IF int(value[ID][3]) is less and equal to 0
        CALL span() from read
        DECLARE continue_loop as False

        ELSE
            CALL available_laptops() from read
```

DECLARE quantity as valid_quantity_for_sell() with parameters as value and ID

DECLARE value[ID][3] as int(value[ID][3]) subtracted by quantity

APPEND ID, quantity in add_to_cart

CALL write_text_file() with parameter as value from write

CALL print_laptop_list() with parameter as value from write

DECLARE additional as True

WHILE additional

INPUT "Do you want to sell more laptops?(Y/N) :" and store in user_input

IF user_input.upper() is equal to N

DECLARE continue_loop as False

DECLARE additional as False

ELIF user_input.upper() is equal to Y

DECLARE continue_loop as True

DECLARE additional as False

ELSE

CALL invalid_input from read

DECLARE additional as True

CALL print()

CALL write_sell_bill with parameter as add_to_cart from write

CALL make_sell from read

DEFINE function valid_id_sell with parameter as value

DECLARE valid_id as False

WHILE not valid_id

```
        TRY
            INITIALIZE ID as int
            IF ID is greater than 0 and ID is less or equal to length of value
                DECLARE valid_id as True
                RETURN ID
            ELSE
                CALL invalid_input from read
        EXCEPT ValueError
            CALL invalid_input from read

DEFINE function valid_id_purchase with parameter as value
    DECLARE valid_id as  False
    WHILE not valid_id
        TRY
            INITIALIZE ID as int
            IF ID is greater than 0 and ID is less or equal to length of value
                DECLARE valid_id as True
                RETURN ID
            ELSE
                CALL invalid_input from read
        EXCEPT ValueError
            CALL invalid_input from read

DEFINE function valid_quantity_purchase with parameter as value
    DECLARE quantity_validity as False
    WHILE not quantity_validity
        TRY
```

```
            INITIALIZE quantity as int
            IF ID is greater than 0
                DECLARE quantity_validity as True
                RETURN quantity
            ELSE
                CALL invalid_input from read
        EXCEPT ValueError
            CALL invalid_input from read

DEFINE function make_purchase()
  DECLARE contents as read_file() from write
  DECLARE value as dictionary() with parameter as contents from write

  DECLARE add_to_cart as list
  DECLARE continue_loop as True
  WHILE continue_loop
      CALL print_laptop_list() with parameter as value from write
      DECLARE ID as valid_id_purchase() with parameter as value
      DECLARE quantity as valid_quantity_for_purchasel() with
      parameter as value which is converted into int
      DECLARE value[ID][3] as int(value[ID][3]) added to quantity
      APPEND ID, quantity in add_to_cart

      CALL write_text_file() with parameter as value from write
      CALL print_laptop_list() with parameter as value from write

          DECLARE additional as True
          WHILE additional is equal to True
```

INPUT "Do you want to buy more laptops?(Y/N) :" and store in user_input

IF user_input.upper() is equal to N

DECLARE continue_loop as False

DECLARE additional as False

ELIF user_input.upper() is equal to Y

DECLARE continue_loop as True

DECLARE additional as False

ELSE

CALL invalid_input from read

DECLARE additional as True

CALL print()

CALL write_purchase_bill with parameter as add_to_cart from write

CALL make_purchase from read

4.3 Pseudo Code for Read

DEFINE function welcome_message()

DISPLAY ("\n--")

DISPLAY (" Welcome to Oasis Laptop Retail ")

DISPLAY ("--\n")

DEFINE function option_selected()

DISPLAY ("Given below are some of the options for you to carry out the needed operations in the system")

DISPLAY ("\n--")

DISPLAY ("Press 1 to purchase a laptop.")

DISPLAY ("Press 2 to sell a laptop.")

DISPLAY ("Press 1 exit.")

DISPLAY ("--\n")

DEFINE function out_of_stock()

DISPLAY ("\n--")

DISPLAY (" The Laptop is OUT OF STOCK!!! ")

DISPLAY ("--\n")

DEFINE function available_laptops()

DISPLAY ("\n--")

DISPLAY (" The Laptop is ACCESSIBLE. ")

DISPLAY ("--\n")

DEFINE function invalid_input()

DISPLAY ("\n--")

DISPLAY (" The Given Data is INVALID!!! ")

```
        DISPLAY ("-----------------------------------------------------------\n")

DEFINE function span()
        DISPLAY ("\n-----------------------------------------------------------")
        DISPLAY ("Sorry!!! The input range is out from our available
        range.")
        DISPLAY ("-----------------------------------------------------------\n")

DEFINE function make_sell()
        DISPLAY ("\n-----------------------------------------------------------")
        DISPLAY ("       The laptop have been SOLD successfully!!!        ")
        DISPLAY ("-----------------------------------------------------------\n")

  DEFINE function make_purchase()
        DISPLAY ("\n-----------------------------------------------------------")
        DISPLAY ("  The laptop have been PURCHASED successfully!!!  ")
        DISPLAY ("-----------------------------------------------------------\n")

  DEFINE function make_purchase()
        DISPLAY ("\n-----------------------------------------------------------")
        DISPLAY (" Thank you for you feedbacks. Please do visit again.  ")
        DISPLAY ("      Have a good day Admin!!!    ")
        DISPLAY ("-----------------------------------------------------------\n")
```

4.4 Pseudo Code for Write

IMPORT read

IMPORT datetime

DEFINE function read_file()

DECLARE file as open("laptop_details.txt", "r")

DECLARE input as file.readlines()

CLOSE file

RETURN input

DEFINE function dictionary() with parameter as content

DECLARE input as dictionary

FOR index up to range as length of content

DECLARE input[index + 1] as content[index] where \n is replaced by “” and split by “,”

RETURN input

DEFINE function print_laptop_list() with parameter as value

PRINT ("----------\n")

PRINT ("ID", "\t", "Laptop Name", "\t", "Brand", "\t", "Price", "\t", "Quantity", "\t", "GEN", "\t", "\t", "CPU")

PRINT ("----------\n")

FOR key, data in value.items()

PRINT (key, "\t", data[0], "\t", data[1], "\t", data[2], "\t", data[3], "\t""\t", data[4], "\t", data[5])

DISPLAY ("\n----------“)

DEFINE function write_text_file() with parameter as value

DECLARE file as open("laptop_details.txt", "w")

FOR data in value.values()

DECLARE write_data as str(data[0]) + "," + str(data[1]) + "," + str(data[2]) + "," + str(data[3]) + "," + str(data[4]) + "," + str(data[5]) + "\n"

CALL file.write with parameter as write_data

CLOSE file

DEFINE function date_and_time()

DECLARE Year as datetime.datetime.now().year

DECLARE Month as datetime.datetime.now().month

DECLARE Day as datetime.datetime.now().day

DECLARE Hour as datetime.datetime.now().hour

DECLARE Minute as datetime.datetime.now().minute

DECLARE Date as (str(Year) + "-" + str(Month) + "-" + str(Day) + " " + str(Hour) + ":" + str(Minute))

RETURN Date

DEFINE function getdate()

DECLARE Year as datetime.datetime.now().year

DECLARE Month as datetime.datetime.now().month

DECLARE Day as datetime.datetime.now().day

DECLARE Date as (str(Year) + "-" + str(Month) + "-" + str(Day))

RETURN Date

DEFINE function write_sell_bill with parameter as add_to_cart

DECLARE contents as read_file()

DECLARE value as dictionary with parameter as contents

DECLARE alphabetic_form as False

WHIILE alphabetic_form is equal False

    DECLARE Customer_Name as String

    IF Customer_Name.isalpha():

        DECLARE alphabetic_form as True

    ELSE

        CALL invalid_input() from read

DECLARE int_contact as False

WHIILE int_contact is equal False

    TRY

        DECLARE Contact as int

        DECLARE int_contact as True

    EXCEPT

        CALL invalid_input from read

PRINT ("\n---------- INVOICE ----------\n")

PRINT ("\n" + "Name: " + Customer_Name)

PRINT ("Phone no.: " + str(Contact))

DECLARE Date as date_and_time()

PRINT ("Sold Date: " + str(Date) + "\n")

PRINT ("--")

PRINT ("ID", "\t", "Customer Name", "\t", "Brand", "\t", "Price", "\t", "Quantity", "\t", "CPU", "\t", "Graphics")

PRINT ("--\n")

```
    DECLARE Total as 0
    FOR index in range(len(add_to_cart))
        DECLARE ID as int(add_to_cart[index][0])
        DECLARE Quantity as int(add_to_cart[index][1])
        DECLARE Name as value[ID][0]
        DECLARE Brand as value[ID][1]
        DECLARE Price as int(value[ID][2].replace("$", "")) * Quantity
        DECLARE CPU as (value[ID][4])
        DECLARE Graphics as (value[ID][5])
        DECLARE Grand_Total as Price * Quantity
        DECLARE Total as old value Total added to Grand_Total

        PRINT (str(index + 1), "\t", Name, "\t", Brand, "\t", str(Price), "\t",
        str(Quantity), "\t", CPU, "\t", Graphics)
        PRINT ("\n")

DECLARE total_price_with_shipping_cost as Total + 100
PRINT ("Grand Total: " + str(Total)+ "\n")
PRINT ("Grand Total with shipping cost: “, str(total_price_with_shipping_cost) +
"\n")

DECLARE file as open(Customer_Name + "_" + str(getdate()) + ".txt", "w")
WRITE in file ("\n----------------------INVOICE----------------------\n")
WRITE in file ("\n" + "Name: " + Customer_Name + "\n")
WRITE in file ("Phone no.: " + str(Contact) + "\n")
DECLARE Date as date_and_time()
WRITE in file ("Date: " + str(Date) + "\n\n")

WRITE in file ("-------------------------------------------------------------")
```

WRITE in file ("\n ID \t Laptop Name \t Brand \t Price \t Quantity \t CPU \t Graphics \n")

WRITE in file ("----------\n\n")

DECLARE Total as 0

FOR index in range(len(add_to_cart))

DECLARE ID as int(add_to_cart[index][0])

DECLARE Quantity as int(add_to_cart[index][1])

DECLARE Name as value[ID][0]

DECLARE Brand as value[ID][1]

DECLARE Price as int(value[ID][2].replace("$", "")) * Quantity

DECLARE CPU as (value[ID][4])

DECLARE Graphics as (value[ID][5])

DECLARE Grand_Total as Price * Quantity

DECLARE Total as old value Total added to Grand_Total

WRITE in file (str(index + 1), "\t", Name, "\t", Brand, "\t", str(Price), "\t", str(Quantity), "\t", CPU, "\t", Graphics)

WRITE in file ("\n\n")

DECLARE total_price_with_shipping_cost as Total + 100

WRITE in file ("----------\n\n")

WRITE in file ("Grand Total: " + str(Total)+ "\n")

WRITE in file ("Grand Total with shipping cost: “, str (total_price_with_shipping_cost) + "\n")

WRITE in file ("\n\n----------")

WRITE in file ("\n Thank you! The laptops have been SOLD successfully. \n")

WRITE in file ("----------")

```
CLOSE file

DEFINE function write_purchase_bill with parameter as add_to_cart
    DECLARE contents as read_file()
    DECLARE value as dictionary with parameter as contents

    DECLARE alphabetic_form as False
    WHIILE alphabetic_form is equal False
        DECLARE Customer_Name as String
        IF Customer_Name.isalpha():
            DECLARE alphabetic_form as True
        ELSE
            CALL invalid_input() from read

    DECLARE int_contact as False
    WHIILE int_contact is equal False
        TRY
            DECLARE Contact as int
            DECLARE int_contact as True
        EXCEPT
            CALL invalid_input from read

    PRINT ("\n---------- INVOICE ----------\n")
    PRINT ("\n" + "Name: " + Customer_Name)
    PRINT ("Phone no.: " + str(Contact))
    DECLARE Date as date_and_time()
    PRINT ("Purchase Date: " + str(Date) + "\n")
```

```
PRINT ("-------------------------------------------------------------------------------------")
PRINT ("ID", "\t", "Customer Name", "\t", "Brand", "\t", "Price", "\t", "Quantity", "\t", "CPU", "\t", "Graphics")
PRINT ("-------------------------------------------------------------------------------------\n")

DECLARE Total as 0
FOR index in range(len(add_to_cart))
    DECLARE ID as int(add_to_cart[index][0])
    DECLARE Quantity as int(add_to_cart[index][1])
    DECLARE Name as value[ID][0]
    DECLARE Brand as value[ID][1]
    DECLARE Price as int(value[ID][2].replace("$", "")) * Quantity
    DECLARE CPU as (value[ID][4])
    DECLARE Graphics as (value[ID][5])
    DECLARE Total as Price * Quantity
    DECLARE Total as old value Total

    PRINT (str(index + 1), "\t", Name, "\t", Brand, "\t", str(Price), "\t", str(Quantity), "\t", CPU, "\t", Graphics)
    PRINT ("\n")

DECLARE Total_VAT as Total + (Total * 13/100)
PRINT (""Total purchase = " + str(Total) + "\n")
PRINT Total purchase with 13% vat = " + str(Total_VAT) + "\n")

DECLARE file as open(Customer_Name + "_" + str(getdate()) + ".txt", "w")
WRITE in file ("\n----------------------INVOICE----------------------\n")
WRITE in file ("\n" + "Name: " + Customer_Name + "\n")
```

WRITE in file ("Phone no.: " + str(Contact) + "\n")

DECLARE Date as date_and_time()

WRITE in file ("Date: " + str(Date) + "\n\n")

WRITE in file ("--")

WRITE in file ("\n ID \t Laptop Name \t Brand \t Price \t Quantity \t CPU \t Graphics \n")

WRITE in file ("--\n\n")

DECLARE Total as 0

FOR index in range(len(add_to_cart))

DECLARE ID as int(add_to_cart[index][0])

DECLARE Quantity as int(add_to_cart[index][1])

DECLARE Name as value[ID][0]

DECLARE Brand as value[ID][1]

DECLARE Price as int(value[ID][2].replace("$", "")) * Quantity

DECLARE CPU as (value[ID][4])

DECLARE Graphics as (value[ID][5])

DECLARE Total as Price * Quantity

DECLARE Total as old value Total

WRITE in file (str(index + 1), "\t", Name, "\t", Brand, "\t", str(Price), "\t", str(Quantity), "\t", CPU, "\t", Graphics)

WRITE in file ("\n\n")

DECLARE Total_VAT as Total + (Total*13/100)

WRITE in file ("--\n\n")

WRITE in file ("Total purchase: " + str(Total)+ "\n")

WRITE in file (("Total purchase with 13% vat = " + str(Total_VAT)+ "\n")

WRITE in file ("\n\n----------")

WRITE in file ("\n Thank you! The laptops have been PURCHASED successfully. \n")

WRITE in file ("----------")

CLOSE file

5. Data Structure

Data Structure is a specific method of storing and organizing data in the computer's memory so that they can be quickly retrieved and effectively used later on as needed. A data structure is a logical or mathematical model for a certain type of data organization. It allows data to be managed in a variety of ways which further increases the efficiency of the program (Anonymous, n.d.)

Python uses different types of data structures and among them there are eight major data structures: Integer, String, Boolean, Float, List, Dictionary, Tuples and Sets.

5.1 Integer

Integer is a whole numbers which has a machine dependent range of values. It is represented by int. It is a whole number, positive or negative, without decimals, of unlimited length.

```
5  def valid_id_for_sell(value):
6      valid_data = False
7      while valid_data == False:
8          try:
9              ID = int(input("Enter the ID of the laptop you want to sell: "))
```

Figure 1 Screenshot of Integer used in program

5.2 String

String is a primitive data type which stores character values. Users cannot modify the contents of the string once it has been created.

```
55 def getdate():
56
57     Year = datetime.datetime.now().year
58     Month = datetime.datetime.now().month
59     Day = datetime.datetime.now().day
60
61     Date = (str(Year) + "-" + str(Month) + "-" + str(Day))
62     return Date
```

Figure 2 Screenshot of String used in program

5.3 Boolean

Boolean represents whether the given expression is true or false. It represents one of the two values. Whenever we compare two values, we can rely on Python to assess the expression and provide its Boolean answer accordingly.

```
84 #Check validity of ID and show messages
85 def valid_id_sell(value):
86 
87     valid_id = False
88     while not valid id:
```

Figure 3 Screenshot of Boolean used in program

5.4 List

List is a data type that store multiple items is just one variable. These are part of the built in data types and are specifically designed for storing collections of data. In order to create a list in Python, we simply use square brackets.

```
44 
45     add_to_cart = []
46     write.print_laptop_list(value)
```

Figure 4 Screenshot of List used in program

5.5 Dictionary

Dictionary is a collection that stores keys and values within a curly bracket. It is ordered, flexible and does not allow repetition. We can change, add or remove items after the dictionary has been created.

```
13 # Function to convert file's content into dictionary
14 def dictionary(content):
15     input = {}
16     for index in range(len(content)):
```

Figure 5 Screenshot of Dictionary used in program

5.6 Float

Float is a data type that can hold number with decimal and fractional part. It can be either positive or negative containing one or more decimals. For example:

x = 20.23

print (x)

5.7 Tuple

Tuples are a type of variable that stores many elements in a single variable. Tuple is one of Python built-in data types for storing data collections. A tuple is an ordered and unchanging collection. For example:

```
demo_tuple = (dell, mac, asus)

print (demo_tuple [1])  #prints mac
```

5.8 Set

Sets are used to hold a number of objects in a single variable. Set is one of Python built-in data types for storing data collections. A set is an unordered as it can appeared in different order every time.

It is immutable once it is created, and unindexed collection. Sets are denoted by curly brackets. For example:

```
demo_set = (dell, mac, asus)          #this is set

print (demo_set)          #prints (dell, mac, asus)
```

6. Program

6.1 Implementation of the program for purchasing laptop

At the beginning of the program, a welcome message is displayed on the main page where the admin is asked to choose between three options i.e. option 1 for purchasing the laptop, option 2 for selling the laptop and option 3 for exiting the program.

```
--------------------------------------------------------------------------------------------------
          Welcome to Oasis Laptop Retail
--------------------------------------------------------------------------------------------------

Given below are some of the options for you to carry out the needed operations in the system

--------------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
--------------------------------------------------------------------------------------------------

Enter an option:
```

Figure 6 Screenshot of main page for purchasing laptop

After the admin chooses option 1, s/he is directed to purchase the laptop. Here, the list of laptops is displayed, and the user is asked to enter the ID and quantity of the laptop that user wants to purchase.

```
Enter an option: 1

--------------------------------------------------------------------------------------------------
ID       Laptop Name     Brand    Price    Quantity        GEN             CPU
--------------------------------------------------------------------------------------------------

1        Razer Blade     Razer    $2000    80              i7 7thGen       GTX3060
2        XPS dell        Dell     $1976    16              i5 9thGen       GTX3070
3        Swift  7        Apple    $900     10              i5 9thGen       GTX3070
4        Macbook Pro16   Apple    $3500    10              i5 9thGen       GTX3070
5        Alien Ware      Alien    $1978    19              i5 9thGen       GTX3070
--------------------------------------------------------------------------------------------------

Enter the ID of the Laptop to purchase: 4
How many laptops do you want to purchase? 1
```

Figure 7 Screenshot of program asking ID and quantity from admin for purchasing laptop

Then a question is asked if the admin wants to purchase more laptops and admin have to choose either Y or N. If we choose Y, it will again continue the same previous steps of asking laptop's ID and quantity and if we choose N, it will terminate and will ask your name and contact. Lastly a bill will be generated.

```
Do you want to buy more laptop?(Y/N) :N

Please Enter the name of the customer: Rashi
Please enter your contact number: 9800000000

--------------------------INVOICE------------------------------

Name: Rashi
Phone no.: 9800000000
Purchase Date: 2023-5-12 1:54
-------------------------------------------------------------------------------------------------
ID       Customer Name   Brand   Price   Quantity        CPU             Graphics
-------------------------------------------------------------------------------------------------

1        Macbook Pro16   Apple   3500    1       i5 9thGen               GTX3070


Total purchase = 7000

Total purchase with 13% vat = 7910.0


-------------------------------------------------------------------------------------------------
          The laptop have been PURCHASED successfully!!!
-------------------------------------------------------------------------------------------------
```

Figure 8 Screenshot of bill generation after purchasing laptop

If the admin wishes to acquire more laptop, see Option 1. It will then display the available stock in the shop and prompt the admin to enter the Laptop Id and the amount of laptops.

```
Given below are some of the options for you to carry out the needed operations in the system

-------------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
-------------------------------------------------------------------------------------------------

Enter an option: 1

-------------------------------------------------------------------------------------------------
ID       Laptop Name     Brand   Price   Quantity        GEN             CPU
-------------------------------------------------------------------------------------------------

1        Razer Blade     Razer   $2000   80              i7 7thGen       GTX3060
2        XPS dell        Dell    $1976   16              i5 9thGen       GTX3070
3        Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4        Macbook Pro16   Apple   $3500   12              i5 9thGen       GTX3070
5        Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
-------------------------------------------------------------------------------------------------

Enter the ID of the Laptop to purchase:
```

Figure 9 Screenshot of available stock after purchasing laptop

6.2 Implementation of the program for selling laptop

At the beginning of the program, a welcome message is displayed on the main page where the admin is asked to choose between three options i.e. option 1 for purchasing the laptop, option 2 for selling the laptop and option 3 for exiting the program.

```
-------------------------------------------------------------------------------------------------------
           Welcome to Oasis Laptop Retail
-------------------------------------------------------------------------------------------------------

Given below are some of the options for you to carry out the needed operations in the system

-------------------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
-------------------------------------------------------------------------------------------------------

Enter an option:
```

Figure 10 Screenshot of main page for selling laptop

After the admin chooses option 2, s/he is directed to purchase the laptop. Here, the list of laptops is displayed, and the user is asked to enter the ID and quantity of the laptop that user wants to sell.

```
Enter an option: 2

-------------------------------------------------------------------------------------
ID       Laptop Name     Brand   Price   Quantity        GEN             CPU
-------------------------------------------------------------------------------------

1        Razer Blade     Razer   $2000   80              i7 7thGen       GTX3060
2        XPS dell        Dell    $1976   16              i5 9thGen       GTX3070
3        Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4        Macbook Pro16   Apple   $3500   12              i5 9thGen       GTX3070
5        Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
-------------------------------------------------------------------------------------

Enter the ID of the Laptop to sell: 2

-------------------------------------------------------------------------------------
  The Laptop is ACCESSIBLE.
-------------------------------------------------------------------------------------

How many laptops do you want to sell? 1
```

Figure 11 Screenshot of program asking ID and quantity from admin for selling laptop

Then a question is asked if the admin wants to sell more laptops and admin have to choose either Y or N. If we choose Y, it will again continue the same previous steps of asking laptop's ID and quantity and if we choose N, it will terminate and will ask your name and contact. Lastly a bill will be generated.

```
Do you want to sell more laptops?(Y/N) :N

Please Enter the name of the customer: Raj
 Please Enter the contact number of the customer: 9812345678

--------------- INVOICE ---------------


Name: Raj
Phone no.: 9812345678
Sold Date: 2023-5-12 2:9


-------------------------------------------------------------------------------
ID       Costumer Name   Brand   Price   Quantity        CPU     Graphics
-------------------------------------------------------------------------------

1        XPS dell        Dell    1976    1       i5 9thGen       GTX3070


Grand Total: 1976

Grand Total with shipping cost:  2076


-------------------------------------------------------------------------------
              The laptop have been SOLD successfully!!!
-------------------------------------------------------------------------------
```

Figure 12 Screenshot of bill generation after selling laptop

If the admin wishes to sell more laptops, see Option 1. It will then display the available stock in the shop and prompt the admin to enter the Laptop Id and the amount of laptops.

```
Given below are some of the options for you to carry out the needed operations in the system

--------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
--------------------------------------------------------------------------------------------

Enter an option: 2

--------------------------------------------------------------------------------------------
ID       Laptop Name     Brand   Price   Quantity        GEN             CPU
--------------------------------------------------------------------------------------------

1        Razer Blade     Razer   $2000   80              i7 7thGen       GTX3060
2        XPS dell        Dell    $1976   15              i5 9thGen       GTX3070
3        Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4        Macbook Pro16   Apple   $3500   12              i5 9thGen       GTX3070
5        Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
--------------------------------------------------------------------------------------------

Enter the ID of the Laptop to sell:
```

Figure 13 Screenshot of available stock after selling laptop

6.3 Implementation of the program to exit

At the beginning of the program, a welcome message is displayed on the main page where the admin is asked to choose between three options i.e. option 1 for purchasing the laptop, option 2 for selling the laptop and option 3 for exiting the program.

```
--------------------------------------------------------------------------------------------------------
           Welcome to Oasis Laptop Retail
--------------------------------------------------------------------------------------------------------

Given below are some of the options for you to carry out the needed operations in the system

--------------------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
--------------------------------------------------------------------------------------------------------

Enter an option:
```

Figure 14 Screenshot of main page

After the admin chooses option 3, the program will terminate with a Thank You message.

```
--------------------------------------------------------------------------------------------------------
           Welcome to Oasis Laptop Retail
--------------------------------------------------------------------------------------------------------

Given below are some of the options for you to carry out the needed operations in the system

--------------------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
--------------------------------------------------------------------------------------------------------

Enter an option: 3

--------------------------------------------------------------------------------------------------------
           Thank you for you feedbacks. Please do visit again.
                      Have a good day Admin!!!
--------------------------------------------------------------------------------------------------------
```

Figure 15 Screenshot of program after choosing option to exit

7. Testing

Test 1: Show implementation of try and except.

Test no:	1
Objective	To provide invalid input and show the message.
Action	Invalid input was entered.
Expected Result	Exception should be handled by the program.
Actual Result	Exception was handled by the program.
Conclusion	The test was successful.

Table 1 Test 1: Implementation of try and except

Output:

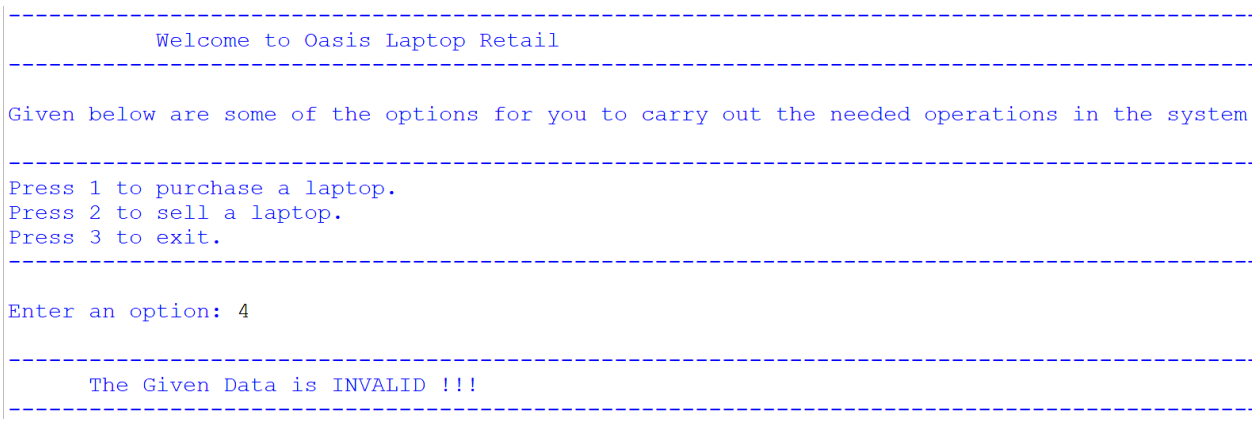


Figure 16 Screenshot of program after entering invalid input

Test 2: Selection purchase and sale of laptops.

Test no:	2
Objective	To provide the negative value or non-existed value as input
Action	➢ Negative value was entered. ➢ Non-existed value was entered.
Expected Result	Invalid message should be shown.
Actual Result	Invalid message was shown.
Conclusion	The test was successful.

Table 2 Test 2: Purchase and sale of laptops.

Output:

```
Enter an option: 1

-------------------------------------------------------------------------------
ID       Laptop Name     Brand   Price   Quantity        GEN             CPU
-------------------------------------------------------------------------------

1        Razer Blade     Razer   $2000   80              i7 7thGen       GTX3060
2        XPS dell        Dell    $1976   16              i5 9thGen       GTX3070
3        Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4        Macbook Pro16   Apple   $3500   10              i5 9thGen       GTX3070
5        Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
-------------------------------------------------------------------------------

Enter the ID of the Laptop to purchase: -8

-------------------------------------------------------------------------------
      The Given Data is INVALID !!!
-------------------------------------------------------------------------------

Enter the ID of the Laptop to purchase: 6

-------------------------------------------------------------------------------
      The Given Data is INVALID !!!
-------------------------------------------------------------------------------

Enter the ID of the Laptop to purchase:
```

Figure 17 Screenshot of program after entering negative and non-existed value

Test 3: File generation of purchase of laptop(s).

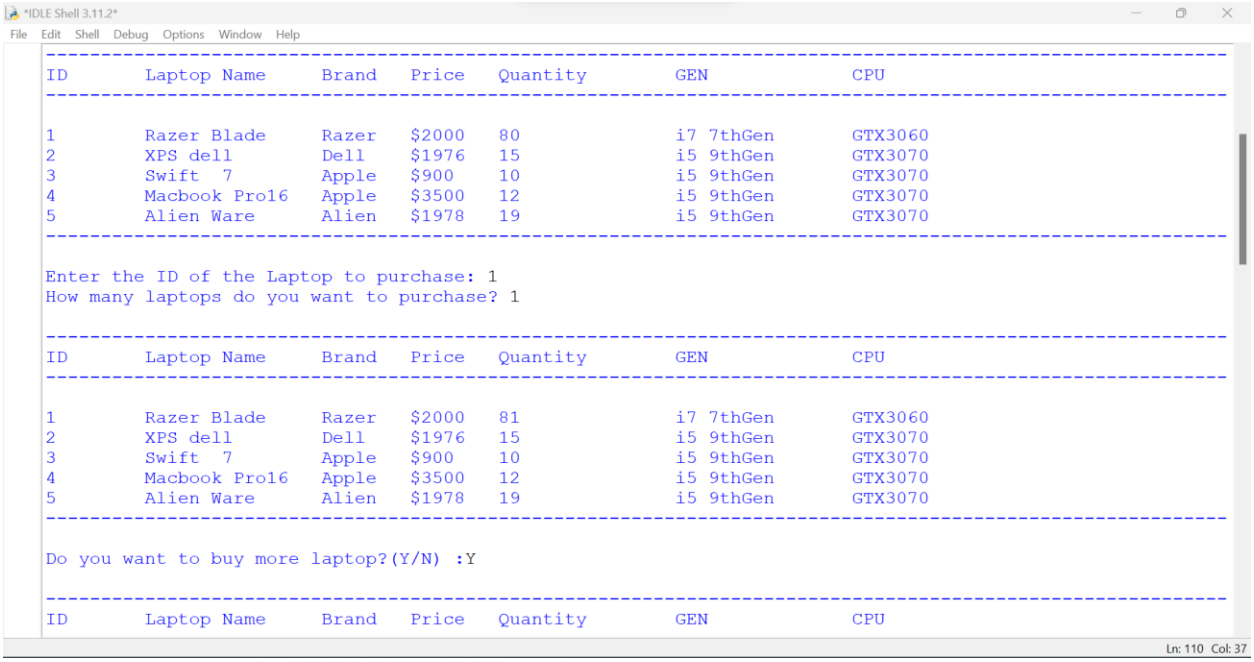
Test no:	3
Objective	To show complete purchase process, output in the shell and the purchased laptops details in a text file.
Action	➢ Purchasing option was selected. ➢ Multiple laptops were purchased. ➢ A bill was displayed in the shell. ➢ A text file of the customer who purchased laptops was created.
Expected Result	A text file should be generated about the purchase of the laptops.
Actual Result	A text file was generated about the purchase of the laptops.
Conclusion	The test was successful.

Table 3 Test 3: Generation of file of purchased laptops.

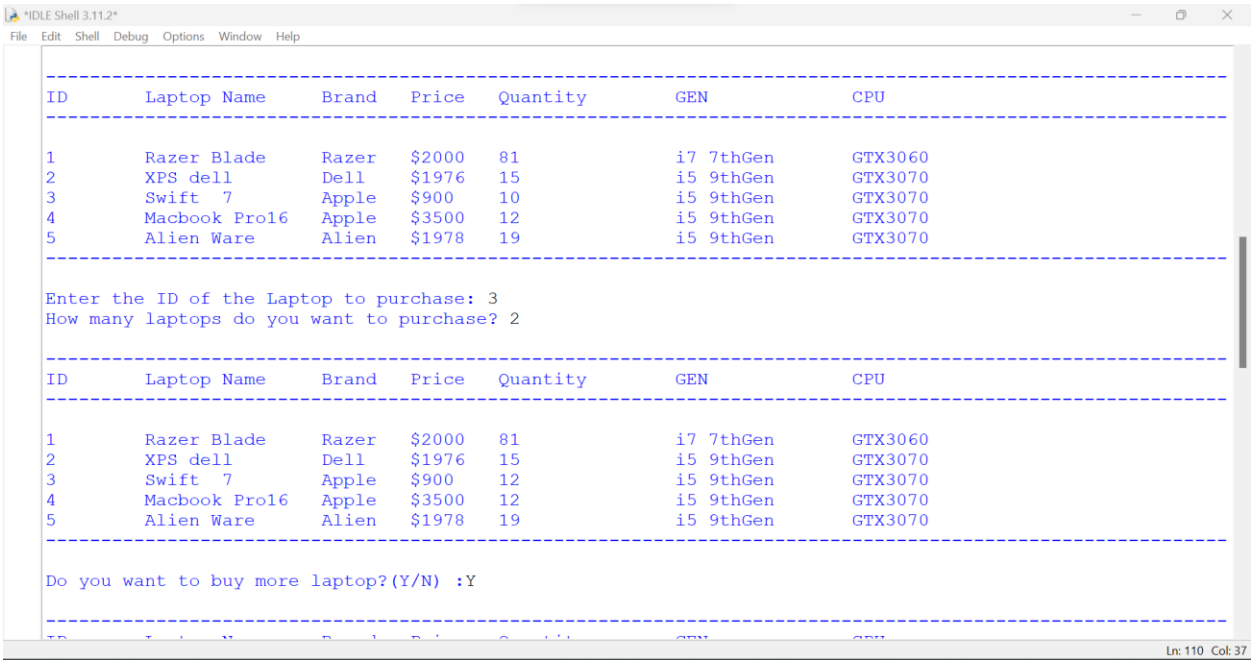
Output:

```
-------------------------------------------------------------------------------------------
          Welcome to Oasis Laptop Retail
-------------------------------------------------------------------------------------------

Given below are some of the options for you to carry out the needed operations in the system

-------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
-------------------------------------------------------------------------------------------

Enter an option: 1
```

Figure 18 Purchase option selected



```
*IDLE Shell 3.11.2*
File  Edit  Shell  Debug  Options  Window  Help
-------------------------------------------------------------------------------------------------------------------------
ID        Laptop Name     Brand   Price   Quantity        GEN             CPU
-------------------------------------------------------------------------------------------------------------------------

1         Razer Blade     Razer   $2000   80              i7 7thGen       GTX3060
2         XPS dell        Dell    $1976   15              i5 9thGen       GTX3070
3         Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4         Macbook Pro16   Apple   $3500   12              i5 9thGen       GTX3070
5         Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
-------------------------------------------------------------------------------------------------------------------------

Enter the ID of the Laptop to purchase: 1
How many laptops do you want to purchase? 1

-------------------------------------------------------------------------------------------------------------------------
ID        Laptop Name     Brand   Price   Quantity        GEN             CPU
-------------------------------------------------------------------------------------------------------------------------

1         Razer Blade     Razer   $2000   81              i7 7thGen       GTX3060
2         XPS dell        Dell    $1976   15              i5 9thGen       GTX3070
3         Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4         Macbook Pro16   Apple   $3500   12              i5 9thGen       GTX3070
5         Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
-------------------------------------------------------------------------------------------------------------------------

Do you want to buy more laptop?(Y/N) :Y

-------------------------------------------------------------------------------------------------------------------------
ID        Laptop Name     Brand   Price   Quantity        GEN             CPU
Ln: 110  Col: 37
```



```
*IDLE Shell 3.11.2*
File  Edit  Shell  Debug  Options  Window  Help

-------------------------------------------------------------------------------------------------------------------------
ID        Laptop Name     Brand   Price   Quantity        GEN             CPU
-------------------------------------------------------------------------------------------------------------------------

1         Razer Blade     Razer   $2000   81              i7 7thGen       GTX3060
2         XPS dell        Dell    $1976   15              i5 9thGen       GTX3070
3         Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4         Macbook Pro16   Apple   $3500   12              i5 9thGen       GTX3070
5         Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
-------------------------------------------------------------------------------------------------------------------------

Enter the ID of the Laptop to purchase: 3
How many laptops do you want to purchase? 2

-------------------------------------------------------------------------------------------------------------------------
ID        Laptop Name     Brand   Price   Quantity        GEN             CPU
-------------------------------------------------------------------------------------------------------------------------

1         Razer Blade     Razer   $2000   81              i7 7thGen       GTX3060
2         XPS dell        Dell    $1976   15              i5 9thGen       GTX3070
3         Swift  7        Apple   $900    12              i5 9thGen       GTX3070
4         Macbook Pro16   Apple   $3500   12              i5 9thGen       GTX3070
5         Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070
-------------------------------------------------------------------------------------------------------------------------

Do you want to buy more laptop?(Y/N) :Y

-------------------------------------------------------------------------------------------------------------------------
Ln: 110  Col: 37
```

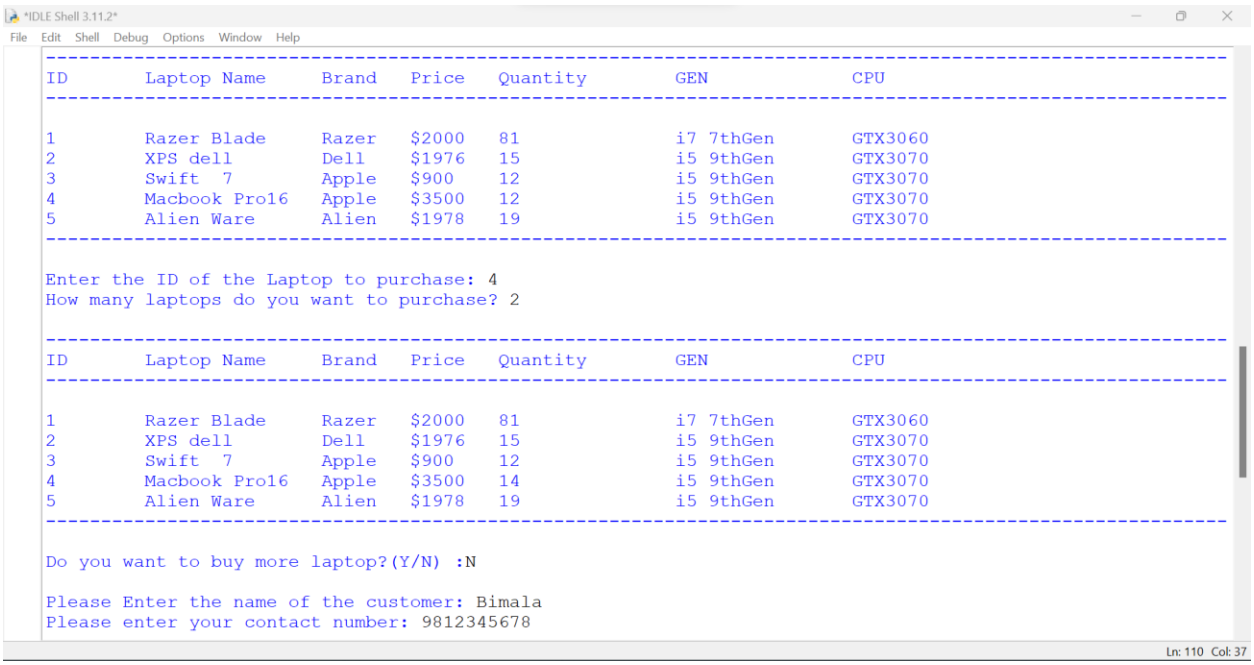


Figure 19 Multiple laptops purchased

```
---------------------------INVOICE-------------------------------

Name: Bimala
Phone no.: 9812345678
Purchase Date: 2023-5-12 3:8
-------------------------------------------------------------------------------
ID       Customer Name   Brand   Price   Quantity       CPU             Graphics
-------------------------------------------------------------------------------

1        Razer Blade     Razer   2000    1       i7 7thGen               GTX3060


2        Swift  7        Apple   1800    2       i5 9thGen               GTX3070


3        Macbook Pro16   Apple   7000    2       i5 9thGen               GTX3070


Total purchase = 28000

Total purchase with 13% vat = 31640.0


-------------------------------------------------------------------------------
          The laptop have been PURCHASED successfully!!!
-------------------------------------------------------------------------------
```

Figure 20 Purchased bill displayed in the shell

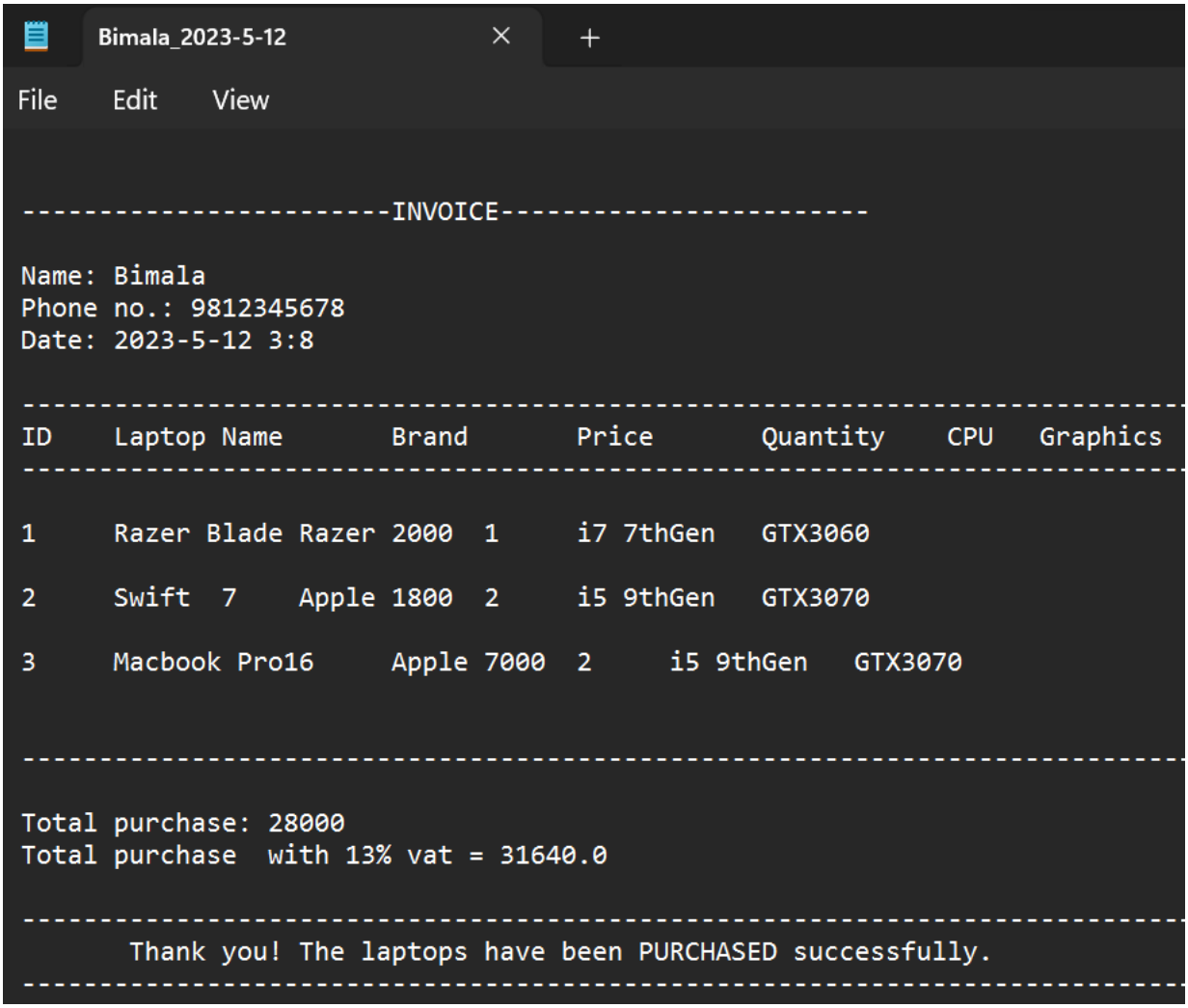


Figure 21 Screenshot of text file of the customer who purchased laptops

Test 4: File generation of sales process of laptop(s).

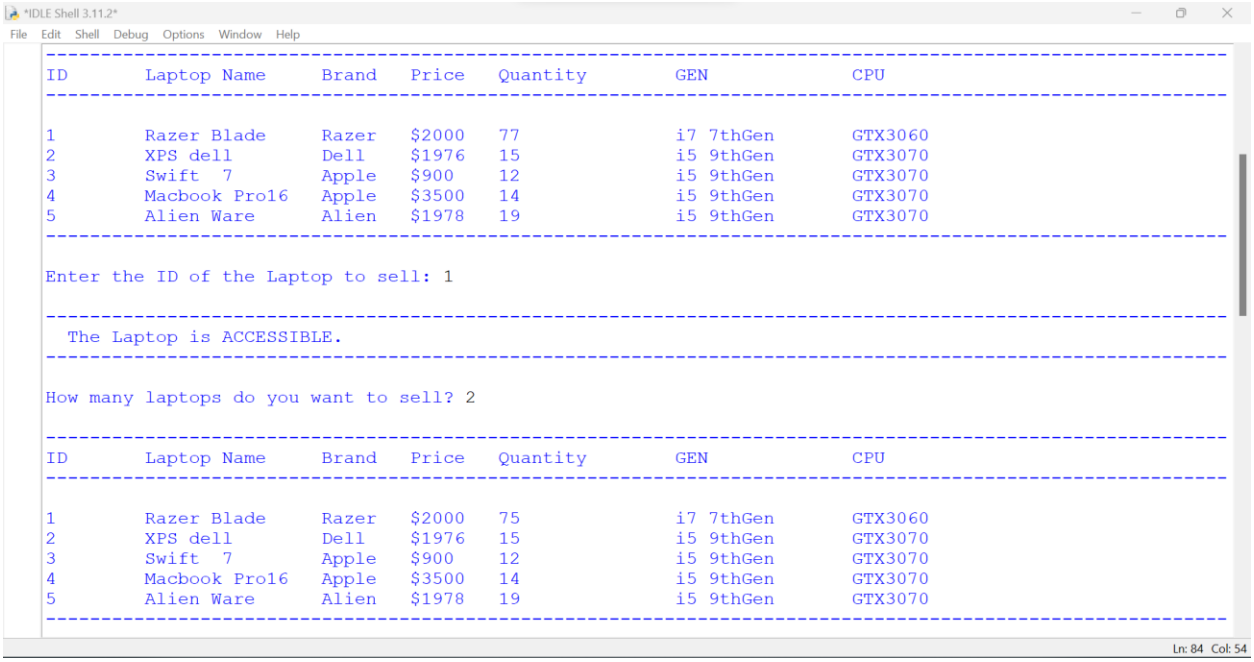
Test no:	4
Objective	To show the complete sales process of the laptop(s), output in the shell and the sold laptop(s) details in text file.
Action	➢ Sales option was selected. ➢ Multiple laptops were sold. ➢ A bill was displayed in the shell. ➢ A text file of the customer who sold laptops was created.
Expected Result	A text file should be generated about the sales of the laptops.
Actual Result	A text file was generated about the sales of the laptops.
Conclusion	The test was successful.

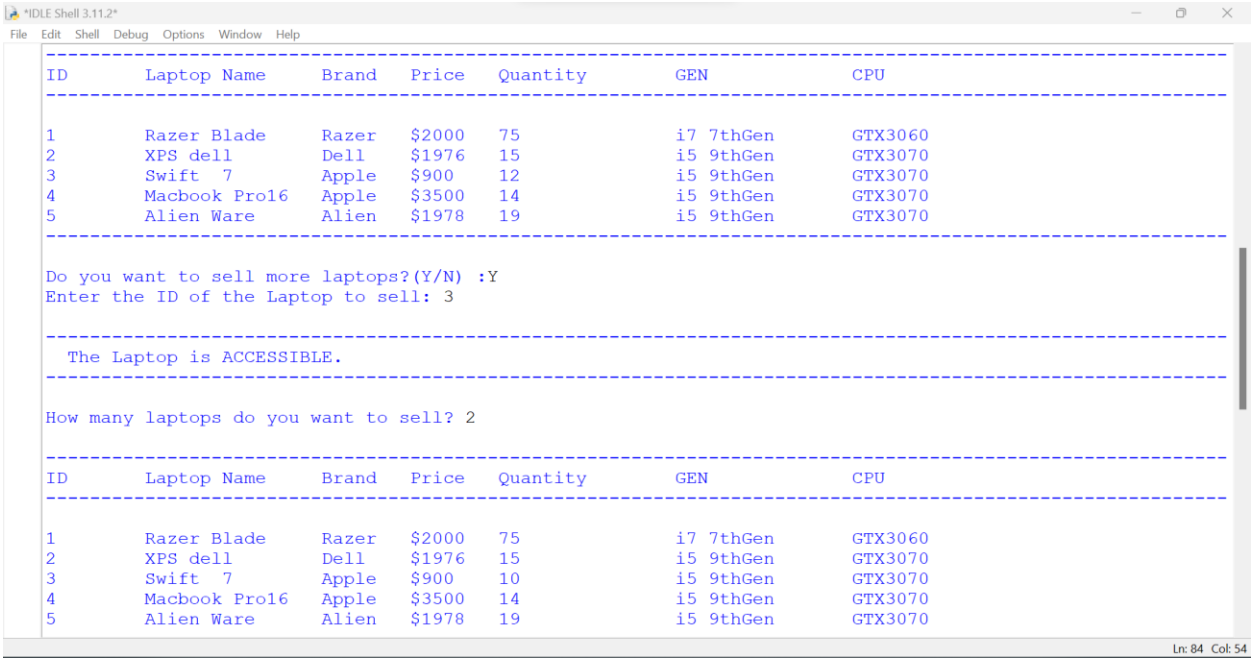
Table 4 Test 4: Generation of file of sales laptops.

Output:

```
-----------------------------------------------------------------------------------------------
          Welcome to Oasis Laptop Retail
-----------------------------------------------------------------------------------------------

Given below are some of the options for you to carry out the needed operations in the system

-----------------------------------------------------------------------------------------------
Press 1 to purchase a laptop.
Press 2 to sell a laptop.
Press 3 to exit.
-----------------------------------------------------------------------------------------------

Enter an option: 2
```

Figure 22 Sales option selected






```
ID       Laptop Name     Brand   Price   Quantity        GEN             CPU
1        Razer Blade     Razer   $2000   75              i7 7thGen       GTX3060
2        XPS dell        Dell    $1976   15              i5 9thGen       GTX3070
3        Swift  7        Apple   $900    10              i5 9thGen       GTX3070
4        Macbook Pro16   Apple   $3500   14              i5 9thGen       GTX3070
5        Alien Ware      Alien   $1978   19              i5 9thGen       GTX3070

Do you want to sell more laptops?(Y/N) :N

Please Enter the name of the customer: Ram
 Please Enter the contact number of the customer: 9812345678
```

Figure 23 Multiple laptops sold

```
--------------- INVOICE ---------------

Name: Ram
Phone no.: 9812345678
Sold Date: 2023-5-12 3:19

-------------------------------------------------------------------------------
ID       Costumer Name    Brand    Price    Quantity        CPU      Graphics
-------------------------------------------------------------------------------

1        Razer Blade      Razer    4000     2       i7 7thGen       GTX3060

2        Swift  7         Apple    1800     2       i5 9thGen       GTX3070

Grand Total: 11600

Grand Total with shipping cost:  11700

-------------------------------------------------------------------------------
              The laptop have been SOLD successfully!!!
-------------------------------------------------------------------------------
```

Figure 24 Sales bill displayed in the shell

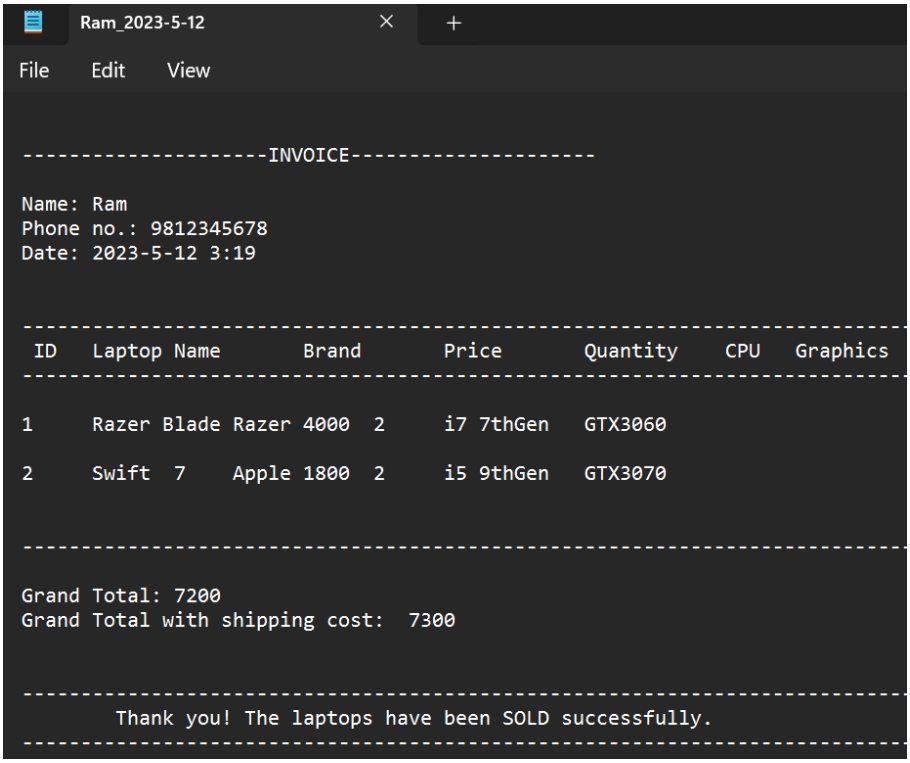


Figure 25 Screenshot of text file of the customer who sold laptops

Test 5: Show the update in stock of laptop(s).

Test no:	5
Objective	To show the quantity being added while purchasing the laptop and quantity being deducted while selling the laptop.
Action	➢ A laptop is purchased. ➢ A laptop is sold.
Expected Result	Stock of laptop should be increased while purchasing and decrease while selling.
Actual Result	Stock of laptop was increased while purchasing and decrease while selling.
Conclusion	The test was successful.

Table 5 Test 5: Update in the stock of laptops

Output:

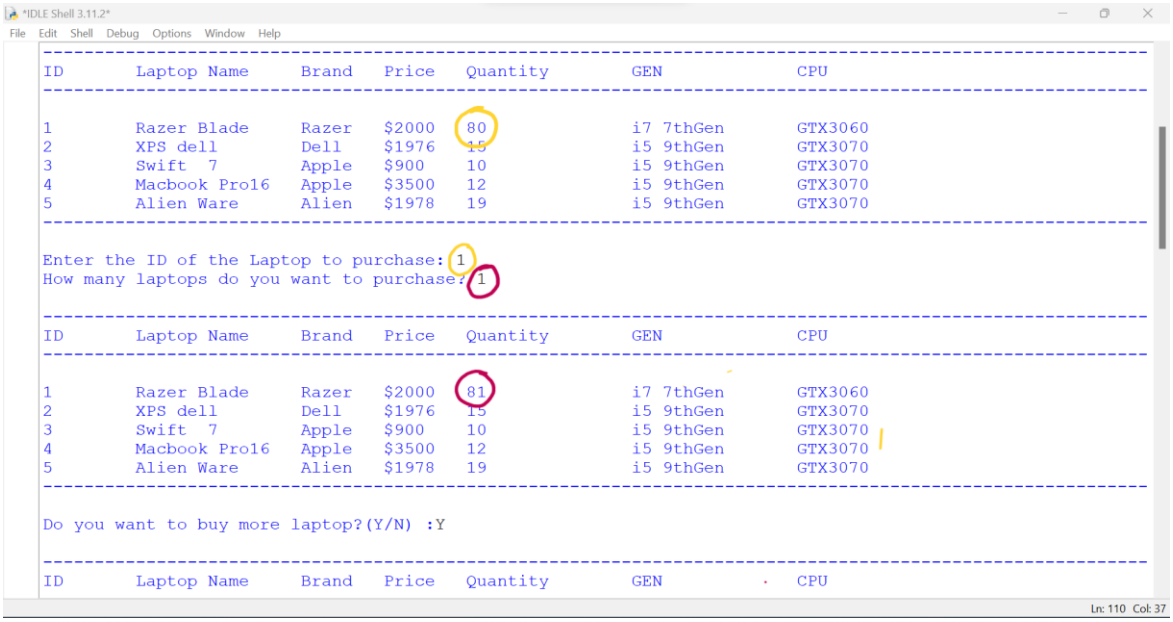


Figure 26 Quantity being added while purchasing the laptop

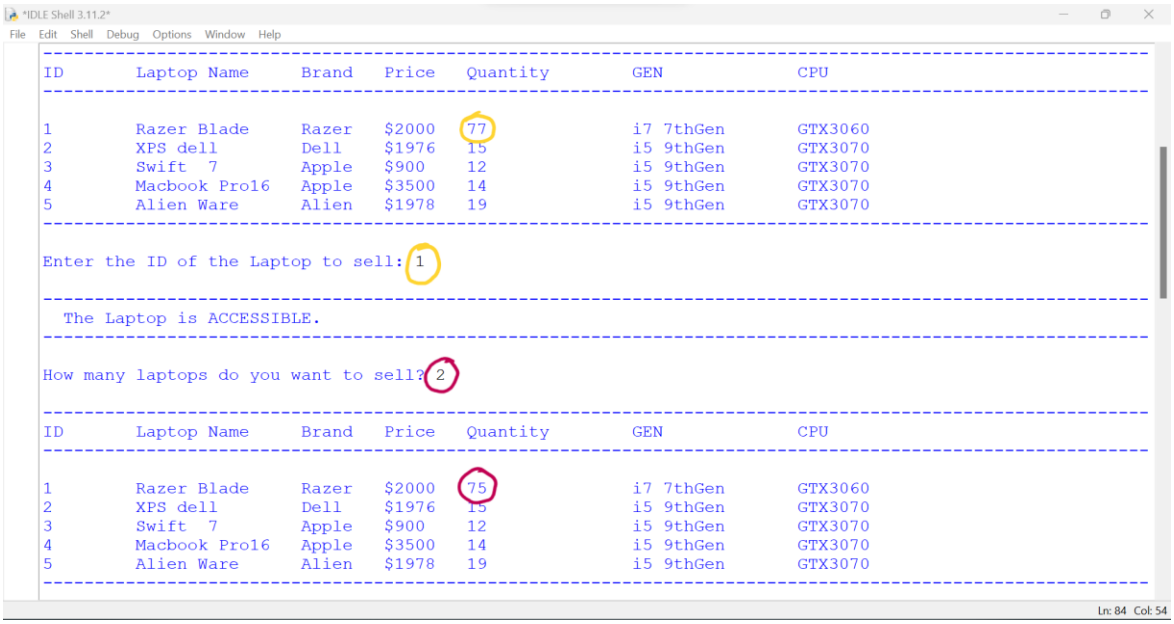


Figure 27 Quantity being deducted while selling the laptop

8. Conclusion

In conclusion, this python project describes how to use Python programming to create an application for a laptop shop including the functionality of purchase and sell. The main.py is created as a super class whereas operation.py, write.py and read.py. The code for this python project was originally written and complied in Virtual Studio Code. In order to simulate program functionality, the project uses several programming techniques, including functions, modules, looping statements, conditional statements, inheritance etc. The project also demonstrates how the simulation was designed and implemented using the Object-Oriented Programming paradigm.

In order to create an effective application, a variety of principles like algorithm, flowchart, pseudo code, data structures and testing must be implemented which we got to learn while doing this coursework. This project taught me the value of modularity and code organization.

Testing and problem fixing were following two crucial project components. We needed to thoroughly test our code to make sure it was operating as intended and to address any errors that we might run across. The hardest part of my coursework was to do coding and implement the coding instructions as this is a new concept to us that gave me the most insight into the subject module. The overall experience has been challenging and rewarding experience.

9. References

Anonymous, n.d. *An Introduction to Data Structures.* [Online]
Available at: https://www.javatpoint.com/data-structure-introduction
[Accessed 24 April 2023].

Karatrantou, A. & Panagiotakopoulos, C., 2008. *Algorithm, Pseudo-Code and Lego Mindstorms Programming,* Venice(Italy): University of Patras.

Rossum, G. V., 2007. *Python (programming language).* [Online]
Available at: http://kelas-karyawan-bali.kurikulum.org/IT/en/2420-2301/Python_3721_kelas-karyawan-bali-kurikulumngetesumum.html
[Accessed 24 April 2023].

Upadhyay, S., 2023. *What Is An Algorithm?.* [Online]
Available at: https://www.simplilearn.com/tutorials/data-structure-tutorial/what-is-an-algorithm
[Accessed 24 April 2023].

10. Appendix

10.1 Code of main.py

```
import operation
import read

# Creation of the main file starts from here
read.welcome_message()
is_running = True

while is_running:
    read.option_selected()

    option_selected = False
    while not option_selected:
        try:
            option = int(input("Enter an option: "))
            option_selected = True
        except:
            read.invalid_input()
            read.option_selected()

    if option == 1:
        operation.make_purchase()

    elif option == 2:
        operation.make_sell()
```

```
    elif option == 3:
        read.display_thanks()
        is_running = False

    else:
        read.invalid_input()
```

10.2 Code of operation.py

```
import read
import write


 #Check the validity of ID and display required message.
def valid_id_for_sell(value):
    valid_data = False
    while valid_data == False:
        try:
            ID = int(input("Enter the ID of the laptop you want to sell: "))


            # ID shouldn't be less than 0 and greater/equal to the length of the dictionary
            if 0 < ID <= len(value):
                if int(value[ID][3]) > 0:
                    valid_data = True
                    return ID
                else:
                    read.out_of_stock()
            else:
                read.invalid_input()
        except:
            read.invalid_input()



#Checks the available quantity of the laptop
def valid_quantity_for_sell(value, ID):
    quantity_validity = False
    while quantity_validity == False:
```

```
        try:
            quantity = int(input("How many laptops do you want to sell? "))
            if quantity > 0 and quantity <= int(value[ID][3]):
                quantity_validity = True
                return quantity
            else:
                read.span()
        except:
            read.invalid_input()



#Function for the sell of laptop
def make_sell():

    contents = write.read_file()
    value = write.dictionary(contents)

    add_to_cart = []
    write.print_laptop_list(value)
    continue_loop = True
    while continue_loop:  # outerloop

        ID = valid_id_sell(value)
        if int(value[ID][3]) <= 0:
            read.span()
            continue_loop = False
        else:
            read.available_laptops()
```

```
            quantity = valid_quantity_for_sell(value, ID)
            value[ID][3] = int(value[ID][3]) - quantity
            add_to_cart.append([ID, quantity])

        write.write_text_file(value)
        write.print_laptop_list(value)

        additional = True
        while additional:  # innerloop
            user_input = input("Do you want to sell more laptops?(Y/N) :")

            if user_input.upper() == "N":
                continue_loop = False
                additional = False

            elif user_input.upper() == "Y":
                continue_loop = True
                additional = False

            else:
                read.invalid_input()
                additional = True

    print()
    # Function for printing and writing the bill
    write.write_sell_bill(add_to_cart)
    read.make_sell()
```

```
#Check validity of ID and show messages
def valid_id_sell(value):

    valid_id = False
    while not valid_id:
        try:
            ID = int(input("Enter the ID of the Laptop to sell: "))
            if ID > 0 and ID <= len(value):
                valid_id = True
                return ID
            else:
                read.invalid_input()
        except ValueError:
            read.invalid_input()

def valid_id_purchase(value):

    valid_id = False
    while not valid_id:
        try:
            ID = int(input("Enter the ID of the Laptop to purchase: "))
            if ID > 0 and ID <= len(value):
                valid_id = True
                return ID
            else:
                read.invalid_input()
        except ValueError:
            read.invalid_input()
```

```
#Checks validity of quantity
def valid_quantity_purchase(value):

    quantity_validity = False
    while not quantity_validity:
        try:
            quantity = int(input("How many laptops do you want to purchase? "))
            if quantity > 0:
                quantity_validity = True
                return quantity
            else:
                read.invalid_input()
        except ValueError:
            read.invalid_input()



#Function for the purchase of the laptop
def make_purchase():

    contents = write.read_file()
    value = write.dictionary(contents)

    add_to_cart = []
    continue_loop = True
    while continue_loop:  # outerloop
        write.print_laptop_list(value)
        ID = valid_id_purchase(value)
```

```
        quantity = int(valid_quantity_purchase(value))
        value[ID][3] = int(value[ID][3]) + quantity
        add_to_cart.append([ID, quantity])

        write.write_text_file(value)
        write.print_laptop_list(value)

        additional = True
        while additional == True:  # inner loop
            user_input = input("Do you want to buy more laptop?(Y/N) :")
            if user_input.upper() == "N":
                continue_loop = False
                additional = False

            elif user_input.upper() == "Y":
                continue_loop = True
                additional = False

            else:
                read.invalid_input()
                additional = True

    print()

     # function to print and write the bill
    write.write_purchase_bill(add_to_cart)
    read.make_purchase()
```

10.3 Code of read.py

```
def welcome_message():
    print("\n-----------------------------------------------------------------------------------------------------------")
    print("          Welcome to Oasis Laptop Retail            ")
    print("-----------------------------------------------------------------------------------------------------------\n")


def option_selected():
    print("Given below are some of the options for you to carry out the needed operations in the system")
    print("\n-----------------------------------------------------------------------------------------------------------")
    print("Press 1 to purchase a laptop.")
    print("Press 2 to sell a laptop.")
    print("Press 3 to exit.")
    print("-----------------------------------------------------------------------------------------------------------\n")


def out_of_stock():
    print("\n-----------------------------------------------------------------------------------------------------------")
    print("      The Laptop is OUT OF STOCK !!!     ")
    print("-----------------------------------------------------------------------------------------------------------\n")


def available_laptops():
```

```
    print("\n-------------------------------------------------------------------------------------------------------
----")
    print("  The Laptop is ACCESSIBLE.    ")
    print("-------------------------------------------------------------------------------------------------------
--\n")



def invalid_input():
    print("\n-------------------------------------------------------------------------------------------------------
----")
    print("      The Given Data is INVALID !!!     ")
    print("-------------------------------------------------------------------------------------------------------
--\n")



def span():
    print("\n-------------------------------------------------------------------------------------------------------
----")
    print(" Sorry!!! The input range is out from our available range.")
    print("-------------------------------------------------------------------------------------------------------
--\n")



def make_sell():
    print("\n-------------------------------------------------------------------------------------------------------
----")
    print("              The laptop have been SOLD successfully!!!              ")
    print("-------------------------------------------------------------------------------------------------------
--\n")
```

```
def make_purchase():
    print("\n-------------------------------------------------------------------------------------------------------")
    print("          The laptop have been PURCHASED successfully!!!          ")
    print("-------------------------------------------------------------------------------------------------------\n")


def display_thanks():
    print("\n-------------------------------------------------------------------------------------------------------")
    print("          Thank you for you feedbacks. Please do visit again.          ")
    print("                    Have a good day Admin!!!                    ")
    print("-------------------------------------------------------------------------------------------------------\n")
```

10.4 Code of write.py

```
import read
import datetime



#Function to call data from the text file
def read_file():
    file = open("laptop_details.txt", "r")
    input = file.readlines()
    file.close()
    return input



# Function to convert file's content into dictionary
def dictionary(content):
    input = {}
    for index in range(len(content)):
        input[index + 1] = content[index].replace("\n", "").split(",")
    return input



#Function to read text file and display list
def print_laptop_list(value):
    print("\n-----------------------------------------------------------------------------------------------------------
----")
    print("ID", "\t", "Laptop Name", "\t", "Brand", "\t", "Price", "\t", "Quantity", "\t", "GEN",
"\t","\t", "CPU")
    print("-----------------------------------------------------------------------------------------------------------
--\n")
```

```
    for key, data in value.items():
        print(key, "\t", data[0], "\t", data[1], "\t", data[2], "\t", data[3], "\t""\t", data[4], "\t",
data[5])


    print("-----------------------------------------------------------------------------------------------------
--\n")



#Function to write and manipulate the text file
def write_text_file(value):
    file = open("laptop_details.txt", "w")
    for data in value.values():
        write_data = str(data[0]) + "," + str(data[1]) + "," + str(data[2]) + "," + str(data[3]) +
"," + str(
            data[4]) + "," + str(data[5]) + "\n"
        file.write(write_data)
    file.close()



#Function to update date and time in bills
def date_and_time():


    Year = datetime.datetime.now().year
    Month = datetime.datetime.now().month
    Day = datetime.datetime.now().day
    Hour = datetime.datetime.now().hour
    Minute = datetime.datetime.now().minute


    Date = (str(Year) + "-" + str(Month) + "-" + str(Day) + " " + str(Hour) + ":" + str(Minute))
```

```
    return Date


def getdate():

    Year = datetime.datetime.now().year
    Month = datetime.datetime.now().month
    Day = datetime.datetime.now().day

    Date = (str(Year) + "-" + str(Month) + "-" + str(Day))
    return Date


#Function to generate bills of sold laptops
def write_sell_bill(add_to_cart):

    contents = read_file()
    value = dictionary(contents)

    alphabetic_form = False
    while alphabetic_form == False:
        Customer_Name = input("Please Enter the name of the customer: ")
        if Customer_Name.isalpha():
            alphabetic_form = True
        else:
            read.invalid_input()

    int_contact = False
```

```
    while int_contact == False:
        try:
            Contact = int(input(" Please Enter the contact number of the customer: "))
            int_contact = True
        except:
            read.invalid_input()


    #Printing of bills
    print("\n---------- INVOICE ----------\n")
    print("\n" + "Name: " + Customer_Name)
    print("Phone no.: " + str(Contact))
    Date = date_and_time()
    print("Sold Date: " + str(Date) + "\n")


    print("\n-------------------------------------------------------------------------------------------------------
----")
    print("ID", "\t", "Customer Name", "\t", "Brand", "\t", "Price", "\t", "Quantity", "\t", "CPU",
"\t", "Graphics")
    print("-------------------------------------------------------------------------------------------------------
--\n")


    Total = 0
    for index in range(len(add_to_cart)):
        ID = int(add_to_cart[index][0])
        Quantity = int(add_to_cart[index][1])
        Name = value[ID][0]
        Brand = value[ID][1]
        Price = int(value[ID][2].replace("$", "")) * Quantity
        CPU = (value[ID][4])
```

```
        Graphics = (value[ID][5])
        Grand_Total = Price * Quantity
        Total += Grand_Total

        print(str(index + 1), "\t", Name, "\t", Brand, "\t", str(Price), "\t", str(Quantity), "\t",
CPU, "\t", Graphics)
        print("\n")

    total_price_with_shipping_cost = Total + 100
    print("Grand Total: " + str(Total)+ "\n")
    print("Grand Total with shipping cost: ", str(total_price_with_shipping_cost)+ "\n")



    #Writing the bills
    file = open(Customer_Name + "_" + str(getdate()) + ".txt", "w")

    file.write("\n----------------------INVOICE----------------------\n")
    file.write("\n" + "Name: " + Customer_Name + "\n")
    file.write("Phone no.: " + str(Contact) + "\n")
    Date = date_and_time()
    file.write("Date: " + str(Date) + "\n\n")

    file.write("\n-------------------------------------------------------------------------------------------
---------")
    file.write("\n ID  \tLaptop Name  \tBrand  \tPrice  \tQuantity  \tCPU  \tGraphics  \n")
    file.write("-------------------------------------------------------------------------------------------
-------\n\n")

    Total = 0
```

```
    for index in range(len(add_to_cart)):
        ID = int(add_to_cart[index][0])
        Quantity = int(add_to_cart[index][1])
        Name = value[ID][0]
        Brand = value[ID][1]
        Price = int(value[ID][2].replace("$", "")) * Quantity
        CPU = (value[ID][4])
        Graphics = (value[ID][5])
        Total = Price * Quantity
        Total += Grand_Total

        file.write(str(index + 1) + "\t" + Name + "\t" + Brand + "\t" + str(Price) + "\t" +
str(Quantity) + "\t" + CPU + "\t" + Graphics)
        file.write("\n\n")

    total_price_with_shipping_cost = Total + 100
    file.write("\n-----------------------------------------------------------------------------------------------
---------\n\n")

    file.write("Grand Total: " + str(Total)+ "\n")
    file.write("Grand Total with shipping cost:  " + str(total_price_with_shipping_cost)+
"\n")
    file.write("\n\n-----------------------------------------------------------------------------------------------
------------")
    file.write("\n        Thank you! The laptops have been SOLD successfully.          \n")
    file.write("-----------------------------------------------------------------------------------------------
-------\n")

    file.close()
```

```
#Function to alter purchase bills
def write_purchase_bill(add_to_cart):

    contents = read_file()
    value = dictionary(contents)

    alphabetic_form = False
    while alphabetic_form == False:
        Customer_Name = input("Please Enter the name of the customer: ")
        if Customer_Name.isalpha():
            alphabetic_form = True
        else:
            read.invalid()

    int_contact = False
    while int_contact == False:
        try:
            Contact = int(input("Please enter your contact number: "))
            int_contact = True
        except:
            read.invalid()

    #Printing Bills
    print("\n-------------------------INVOICE-----------------------------")
    print("\n" + "Name: " + Customer_Name)
    print("Phone no.: " + str(Contact))
    Date = date_and_time()
    print("Purchase Date: " + str(Date))
```

```
    print("-------------------------------------------------------------------------------------------------------
--")
    print("ID", "\t", "Customer Name", "\t", "Brand", "\t", "Price", "\t", "Quantity", "\t", "CPU",
"\t\t", "Graphics")
    print("-------------------------------------------------------------------------------\n")


    Total = 0
    for index in range(len(add_to_cart)):
        ID = int(add_to_cart[index][0])
        Quantity = int(add_to_cart[index][1])
        Name = value[ID][0]
        Brand = value[ID][1]
        Price = int(value[ID][2].replace("$", "")) * Quantity
        CPU = (value[ID][4])
        Graphics = (value[ID][5])
        Total = Price * Quantity
        Total += Total


        print(str(index + 1), "\t", Name, "\t", Brand, "\t", str(Price), "\t", str(Quantity), "\t",
CPU, "\t\t", Graphics)
        print("\n")


    Total_VAT = (Total + (Total * 13/100))
    print("Total purchase = " + str(Total) + "\n")
    print("Total purchase with 13% vat = " + str(Total_VAT) + "\n")


    # bill generation(writing bill) starts here
    file = open(Customer_Name + "_" + str(getdate()) + ".txt", "w")  # a text file with the
name of the user is created
```

```
    file.write("\n----------------------INVOICE----------------------\n")
    file.write("\n" + "Name: " + Customer_Name + "\n")
    file.write("Phone no.: " + str(Contact) + "\n")
    Date = date_and_time()
    file.write("Date: " + str(Date) + "\n\n")

    file.write("-------------------------------------------------------------------------------------------------------")
    file.write("\nID  \tLaptop Name  \tBrand  \tPrice  \tQuantity  \tCPU  \tGraphics  \n")
    file.write("-------------------------------------------------------------------------------------------------------\n\n")

    Total = 0
    for index in range(len(add_to_cart)):
        ID = int(add_to_cart[index][0])
        Quantity = int(add_to_cart[index][1])
        Name = value[ID][0]
        Brand = value[ID][1]
        Price = int(value[ID][2].replace("$", "")) * Quantity
        CPU = (value[ID][4])
        Graphics = (value[ID][5])
        Total = Price * Quantity
        Total += Total

        file.write(str(index + 1) + "\t" + Name + "\t" + Brand + "\t" + str(Price) + "\t" + str(Quantity) + "\t" + CPU + "\t" + Graphics)
        file.write("\n\n")
```

```
    Total_VAT = (Total + (Total*13/100))

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

    file.write("Total purchase: " + str(Total)+ "\n")

    file.write("Total purchase  with 13% vat = " + str(Total_VAT)+ "\n")

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

    file.write("\n       Thank you! The laptops have been PURCHASED successfully.    \n")

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


    file.close()
```