Python work

# INTRODUCTION

This is the second semester coursework for Fundamentals of Computing, written using the python programming language. This coursework has a specific set of goals which I believe has been met by the program developed by me. Before diving into the goals and other details about the programing language, lets discuss about what python is and the other tools used for developing the program.

* Python : Python is a high level, interpreted programming language. It can follow both object oriented and functional paradigms and is dynamically typed i.e., data types and function return types do not have to be pre-defined. Python has very easy to read and understand syntax, which makes it easy for beginners to learn and makes it attractive for rapid application development (Python.org, 2023). This program was developed in python3 which is the latest stable version.
* IDLE: IDLE is python’s Integrated Development and Learning Environment. It is installed by default when installing python in windows. This program was written, run, and debugged inside python IDLE. It has features like syntax highlighting, auto indentation to some extent and more. It also allows the user to execute simple lines of python statements (Python.org, 2023).
* Ms Word: MS-Word is a word processing software developed by the Microsoft Corporation (Byju's, 2020). It was used to develop this report.
* Draw.io : Draw.io is a website used to create UML(Unified Modal Language) diagrams. This website was used to create the class diagrams for all four classes.

The goals of this coursework are straightforward. We have been instructed to develop a python program that resembles point of sales software in the real world. The program must ask the user (the store) if they want to sell laptop to consumers, buy laptop from distributors or exit the program. Incase of selling laptop, the program must display the available stock, validate the id chosen by the user and validate the stock. When the selling process is complete, a proper bill must be generated and the stock inside the txt file must also be updated. A summary of the bill should be shown in the python shell.

When the option to buy laptop from distributors is selected, the program must increase the stock of the selected laptop, with the same validations as the one in the sell laptop to consumer. A bill must also be generated with the required information.

The generated program meets all the requirements along with some extras like an option to display the current stock and an option to add laptop not currently in stock.

The program works fully inside the IDLE shell and is closed only when the user instructs it to.

# Discussion and Analysis

This section contains the algorithm, flowchart and pseudocode of the python program.

## Algorithm

An algorithm is the step-by-step process that needs to be followed in order to produce the desired result. Algorithms are not constrained within any programming language, rather they are only the procedures that need to be followed and can be implemented in any language desired by the user (Upadhyay, 2023).

* **STEP 0: START**
* **STEP 1:** START function welcomeScreenMain
* **STEP 2:** Import the required modules
* **STEP 3:** Display the welcome screen
* **STEP 4:** Take user input for the required option
* **STEP 5:** validate the user input
* **STEP 6:** If user input is 1 call the display method and sellLaptopMain method
  + **STEP 6.1**: Display the current stock of the laptop
  + **STEP 6.2:** Take user input for which laptop to buy and validate it
  + **STEP 6.3:** Ask user for quantity to buy and validate it
  + **STEP 6.4:** Confirm user selection
  + **STEP 6.5:** Decrease the stock from the txt file
  + **STEP 6.6:** Ask use if they want to buy more, repeat from 6.1. if yes
  + **STEP 6.7:** Generate a bill for the purchase
  + **STEP 6.8:** Generate a purchase summary inside the shell
* **STEP 7:** if user input is 2 call the display method and purchaseLaptopMain method
  + **STEP 7.1**: Display the current stock of the laptop
  + **STEP 7.2:** Get user input for which sku to add
  + **STEP 7.3:** If the sku already exists, ask directly for quantity to add
  + **STEP 7.4:** Else ask user for the laptop details
  + **STEP 7.5:** Call the update stock method and update the txt file
  + **STEP 7.6:** Ask the user if they want to buy more, if buy more is yes then repeat from step 7.1
  + **STEP 7.7:** Else call the generate bill method and print the bill summary
* **STEP 8:** if user input is 3 call the display method
* **STEP 9:** Else exit the program
* **STEP 10: END**

## Flow Chart

Flowchart graphically illustrates the step, sequences, and decisions of a workflow (Tesm Asana, 2023).

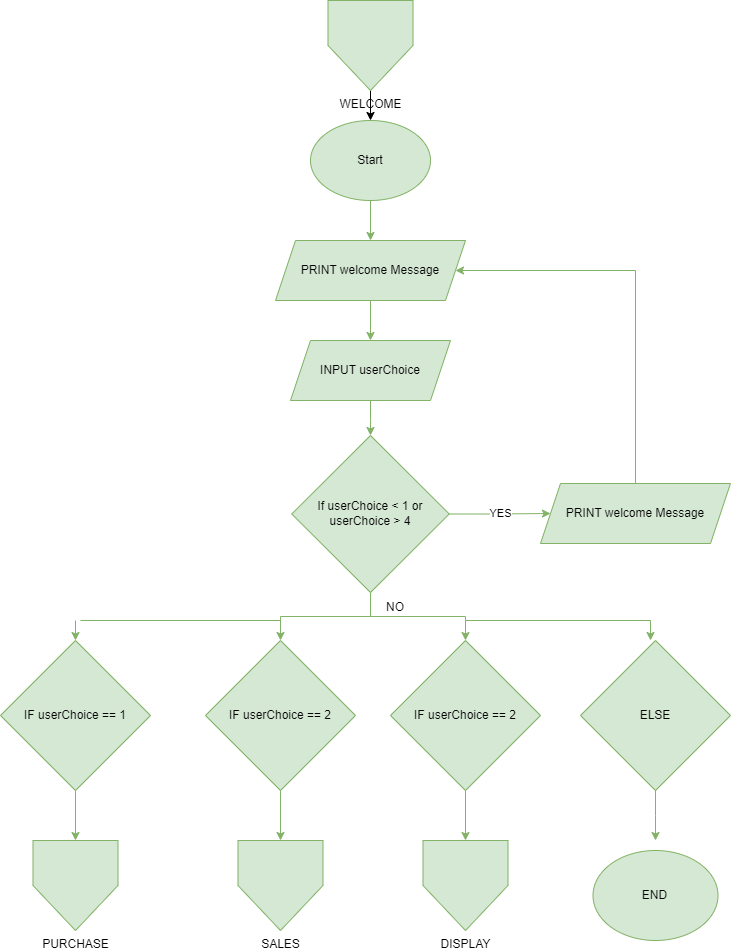


Figure : Welcome Flowchart

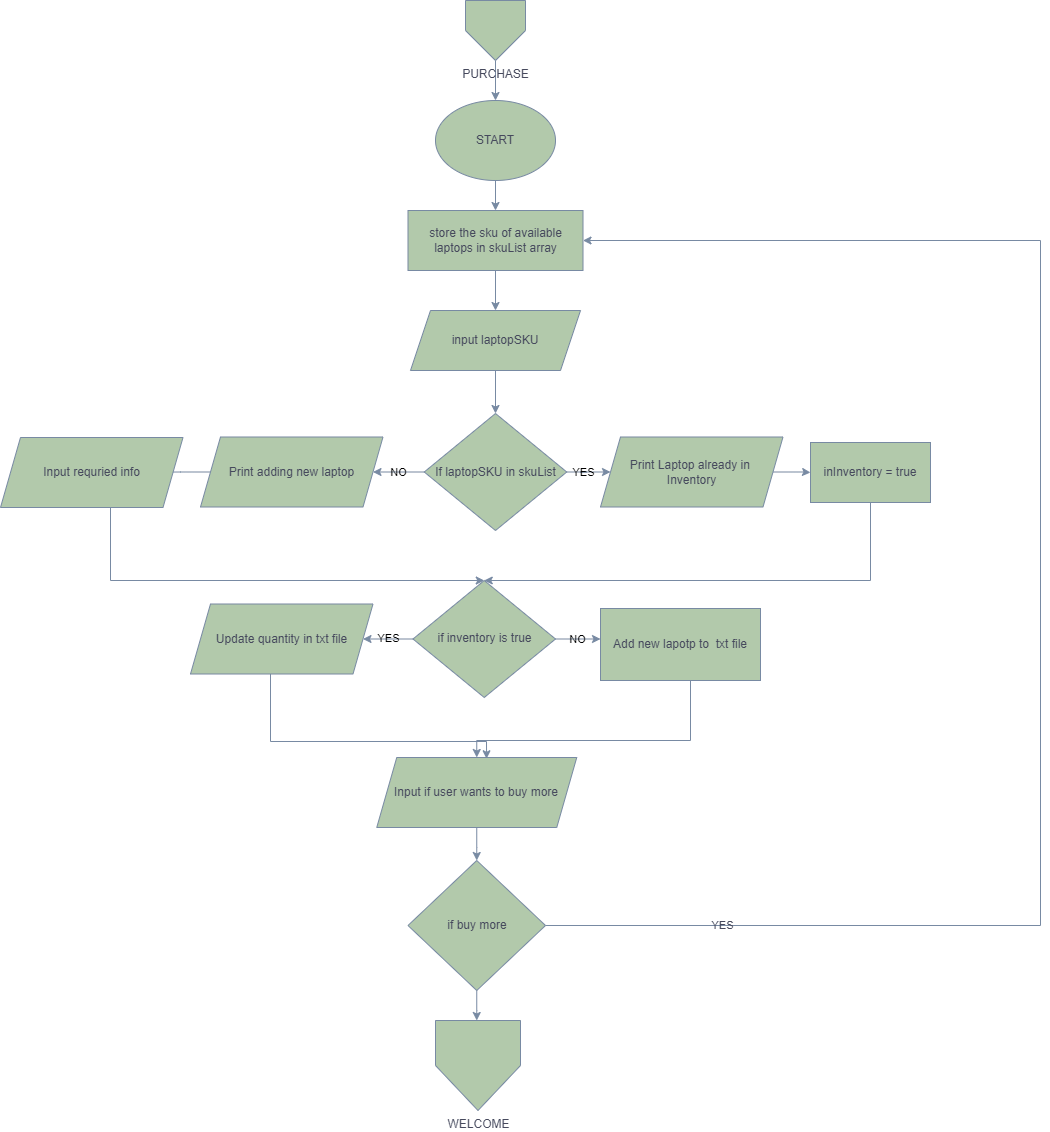


Figure : Purchase Flowchart

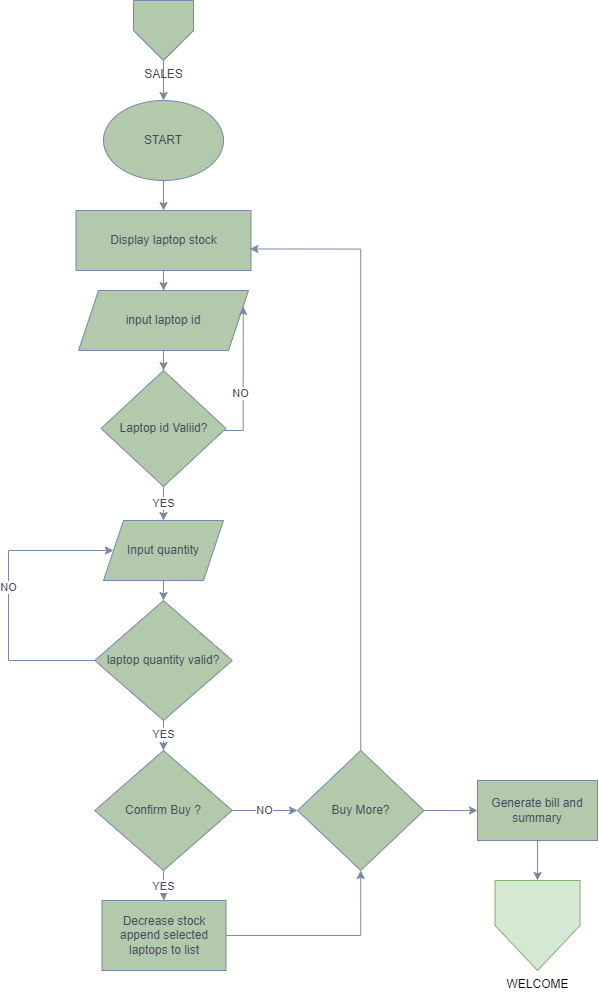


Figure : Sell Flowchart

A diagram of a string

Description automatically generated with medium confidence

Figure : Display Flowchart

## Pseudo Code

Pseudo code is an informal way of writing code in high level human understandable language. In simple words it means to write code in plain English, that can be understood by anyone, even a non tech person (S, 2023).

1. Main.py pseudocode

This is my pseudocode

1. Operations.py pseudocode

This is my pseudocode

1. sellLaptop.py pseudocode

This is my pseudocode

1. purchaseLaptop.py pseudocode

This is my pseudocode

## Data Structures

Data structures as its name suggests is the way in which data is stored and organized (Geeksforgeeks.org, 2023). There are various types of data structures in python, some of them are discussed below.

1. Lists :

Lists in python is an ordered collection of data. Since python is dynamically typed, lists in python are very flexible as it can store data of more than one type. Elements in lists are accessed through their index. The very first element is 0 and it increases as new elements are added. Lists are mutable, i.e. the data can be changed later on the code (Geeksforgeeks.org, 2023).

A list can be one dimensional or two dimensional. Two-dimensional list is a normal list with other lists stored inside. We will discuss about 2D lists in detail later as it is the data structure I have implemented in my code.

1. Dictionaries

Dictionaries in python store data in Key-Value pairs. Every key has its own values. Unlike in list where values are accessed using indexes, we use the key to access the values in dictionaries. Dictionaries in python are enclosed in two curly brackets. Like lists, dictionaries have their own unique methods for manipulating the elements (Akash, 2019).

1. Tuples

Tuples are similar to lists, the only difference being that tuples unlike lists are not mutable i.e. the data inside the tuples cannot be changed. Tuples in python are declared and used by enclosing the data inside small braces (Akash, 2019).

1. Sets

Sets in python is a collection unordered unique element. I.e., even if an element is repeated more than once, it will be registered inside a set only once. Sets in python have the same logical operation as sets in real life (Akash, 2019).

* **Data structure used in this program**

I decided to use 2D lists to manipulate data in this program. I was debating if I should use dictionaries or 2d lists for the various functionalities in this program and at last decided I would use 2d lists.

I used 2d lists in this python program because I found It easier to traverse through than dictionaries. For me it was easier to access individual elements inside a list, than inside dictionaries. Using indexes to access items and mutate items was much easier for me than using keys to access values. I do think that dictionaries provide more robust validation of existing stock and make it easier for the program to avoid any kind of duplication as keys in a dictionary cannot be repeated and must be unique. But for me personally the deeper understanding of list I had made me weigh in on using lists as the data structure for my program more than using dictionaries.

If I were to do this project again, I think I would use dictionaries rather than lists for the sole reason that I would make identifying already existing stock much easier with the unique key value system. I have overcome that validation barrier in my program by adding a unique SKU for each laptop which the user has to enter before purchasing. But my program doesn’t account for the fact that users might sometimes make a mistake and enter the wrong sku.

The screenshot of the code containing the function which stores the current quantity of laptop inside a 2d list is given below.

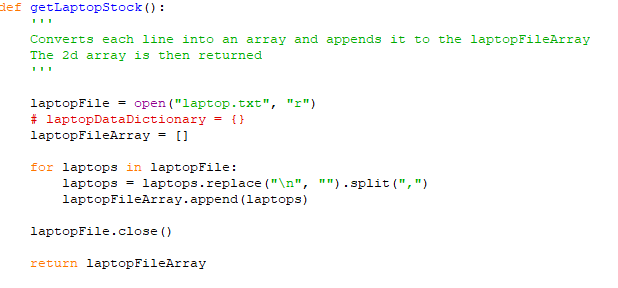


Figure : Implementation of 2d list

This function first opens the txt file for reading. It then iterates over the lines in the file and replaces the new line character with an empty character. Then it splits the information after each comma and returns a list. That list is then stored in another list called the laptopFileArray. This list is then iterated over as required by other functions.

# Program

This portion explains the whole implementation of the program

The program first greets the user with a welcome screen. There are four options provided to the user on this screen. Sell laptops to consumers, buy laptop from distributors, display the current stock of laptops available and exit.

There are four modules in this program, main.py, operations.py, sellLaptop.py and purchaseLaptop.py. The main.py program contains only the welcome screen, and it redirects the user to the correct function and module as per their choice. Module is inside a while loop which means that the program will continue to run indefinitely until the users exits.

The sellLaptop.py module handles all the things related to the selling of laptop to the consumers. It first asks which laptop the user want to sell, if the laptop id is invalid it shows an error and asks for input again. When the suitable value is entered it again asks the user the quantity of laptop they want to buy. This input is also invalidated and asked again if the quantity is higher than the stock, doesn’t exist or is an invalid value. The user is asked to confirm their purchase for one last time. If they confirm the purchase the program continues running, else the program is cancelled and the main.py is run again. The decreaseStock function is called now which as its name suggests decreases the quantity of laptop bought by the user from the txt file.The user is asked if they want to buy more, if they say yes the program loops again, else the user is asked if he or she wants shipping along with their other details. If the User wants shipping the value if shipping is set to 500, else 0. At last an appropriate bill is generated for sales and a summary is printed on the shell.

If the user chooses the purchase from distributor option, first a list of available laptops is printed. Then the required items like the current stock list, sku list, and other details are initialized. The program then asks the user to enter the sku of the laptop they want to buy. It then checks if the sku user entered is in the lists of sku of the current stock. If the sku is already present, the index of that sku is stored and the user is directly asked for quantity, skipping the details. IF the sku doesn’t exist, the program asks the user for all the necessary details to add a new laptop. If the laptop already exists In the list, only the quantity of the laptop is updated. Else a new string is initialized containing all the required information and is appended to the end of the txt file. Then the user is asked if they would like to buy more, if they say yes this function is repeated again. Else, a bill is generated with the necessary information and a summary of purchase if printed on the shell.

If the user chooses option three which is for displaying the laptop stock, the display function from the operations module is called.

* Complete process of selling a laptop

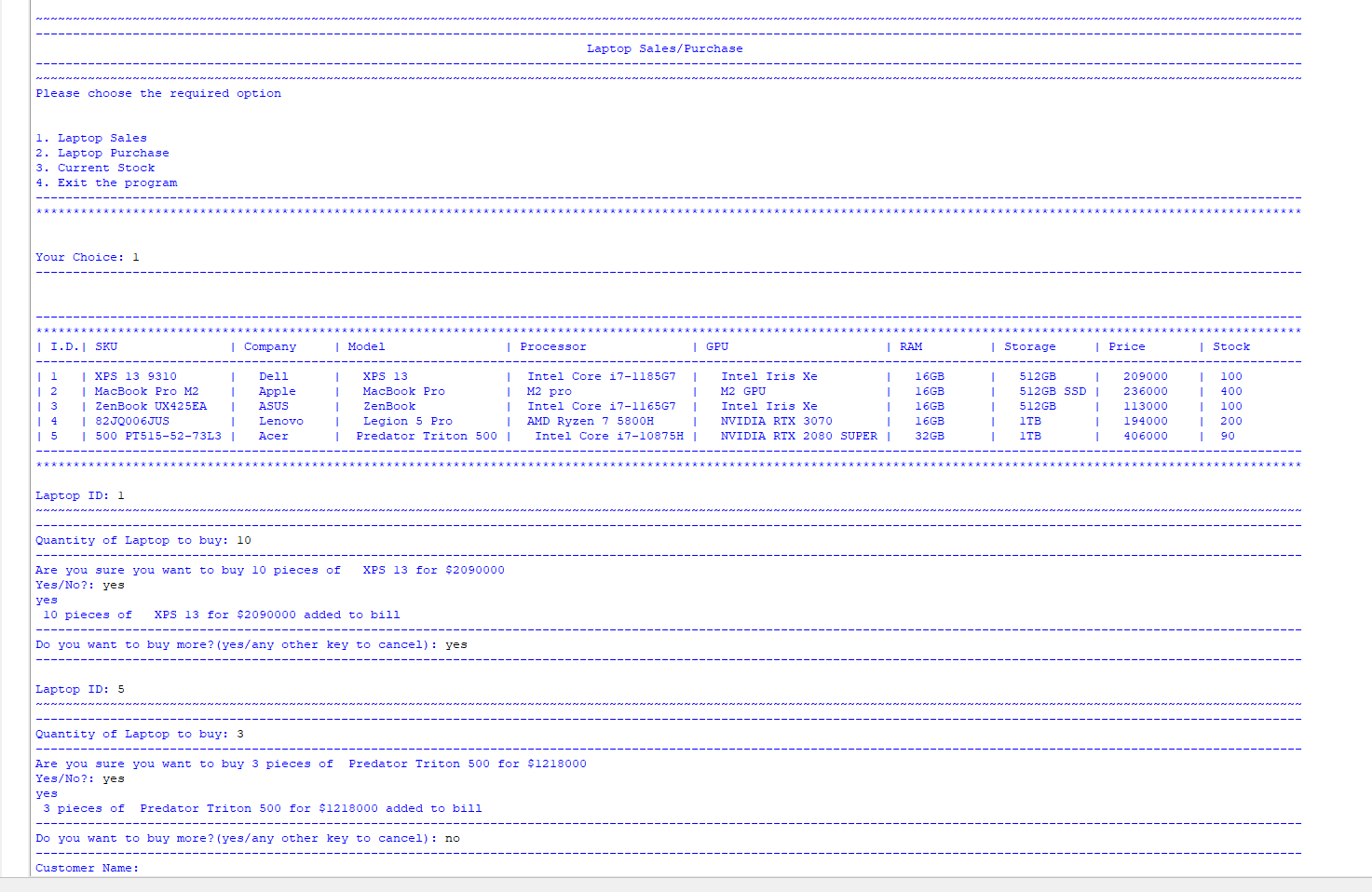


Figure : Selling 2 laptops

A screenshot of a computer

Description automatically generated with medium confidence

Figure : bill generation after Sales

* Complete process of buying a laptop

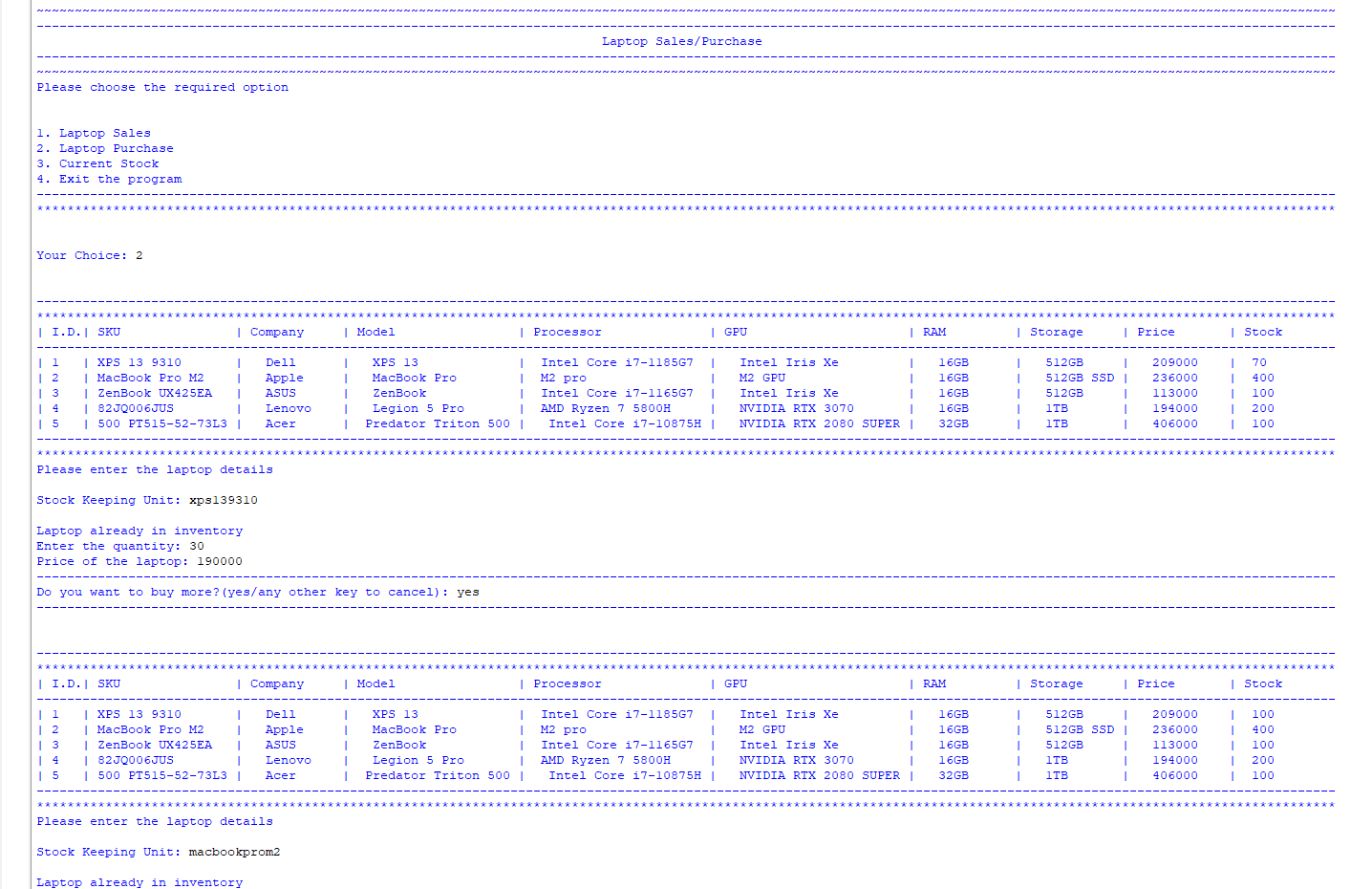


Figure : purchasing 2 from distributors

A picture containing text, line, font, screenshot

Description automatically generated

Figure : Purchase bill generation

* Bill being generated after sales and purchase

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Bill generation

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Consumer bill after laptop sales

A picture containing screenshot, text, line

Description automatically generated

Figure : Supplier bill after laptop purchase

* Process of exiting the program

A picture containing text, line, font, screenshot

Description automatically generated

Figure : Process of exiting the program

# Testing

Testing is a critical part of software development. It helps find bugs and edge case early on before the program is deployed to users. This streamlines the user experience and lessens post deployment maintenance. Testing is the process of evaluating and verifying if a program works as it’s supposed to (IBM, 2016).

## Test 1: Implementation of Try and Except

|  |  |
| --- | --- |
| Test No: | 1 |
| Objective: | To show the implementation of Try and Except |
| Action: | * Run the main.py modue * Enter a string value instead of number in input |
| Expected Result: | The program should show invalid input in the shell and ask for the user input again |
| Actual Result: | The program showed an error and asked for user input again |
| Conclusion: | The test is successful |

Table : Test -1

Try and Except block has been used throughout the code for validation various numeric input. This is an example of try and catch on the main welcome screen where a user must choose what they wish to do.

A picture containing text, screenshot, font, line

Description automatically generated

Figure : Test 1 - invalid input

As seen in the screenshot above, when “invalid input" is input on the your choice field instead of a numerical value, instead of an error stopping the program, the catch block prints that the user choice is invalid and asks for the user choice again,

## Test 2: Provide negative and non-existent value in purchase and sales laptop index

|  |  |
| --- | --- |
| Test No: | 2 |
| Objective: | To check how the program handles non-existent and negative value for purchase and sales index |
| Action: | * Provide negative input in sales index * Provide non-existent value in sales index * Provide negative input in purchase index * Provide non-existent value in purchase index |
| Expected Result: | All the invalid input must be handled correctly, and the program must ask for input again |
| Actual Result: | The invalid input was handled correctly and the program asked for input again |
| Conclusion: | The test is successful |

Table : Test -2

Unlike non-numeric values, negative and non-existent values don’t cause an exception on the user input part. It will cause index out of range exception later on throughout the code, so it must be validated before being used. We use a if else block inside a while loop to achieve this.

A close-up of a computer screen

Description automatically generated with low confidence

Figure : Test 2 - negative and invalid input for sales

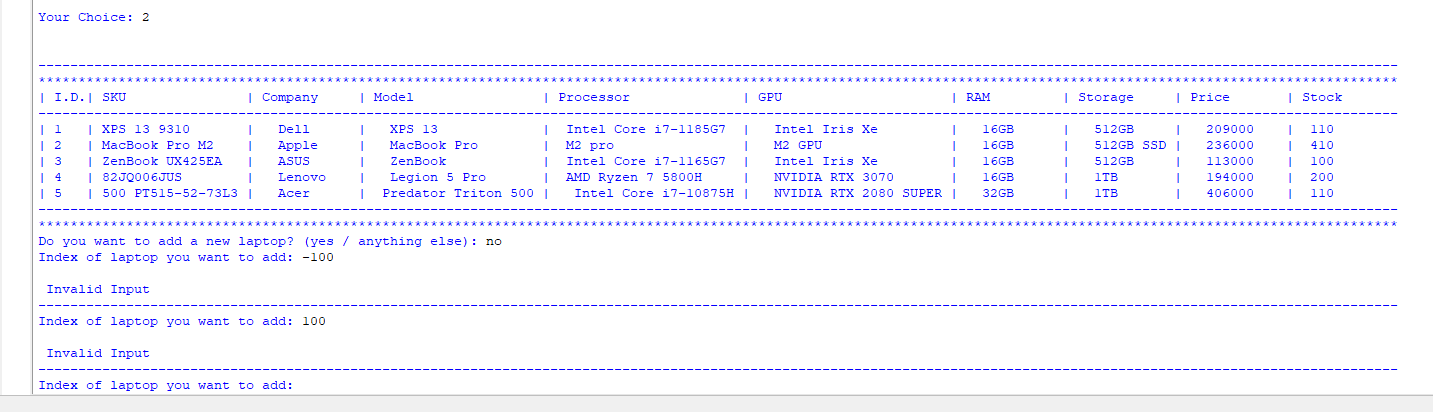


Figure : Test 2 - negative and invalid input for purchase

As seen in the above two screenshots, when an invalid or non-existent value is entered, the program validates it and keeps on asking for user input until the correct value is given

## Test 3: File generation of laptop purchase with multiple laptops

|  |  |
| --- | --- |
| Test No: | 3 |
| Objective: | To check file generation for purchase of multiple laptops |
| Action: | * Purchase multiple laptops * Check purchase summary in shell * Check .txt bill |
| Expected Result: | The information of multiple purchases must be shown correctly with vat and other required info |
| Actual Result: | All the information was correctly shown |
| Conclusion: | The test is successful |

Table : Test -3

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Test -3 purchasing multiple

A screenshot of a computer

Description automatically generated with low confidence

Figure : Test -3 Purchase Summary

A screenshot of a computer program

Description automatically generated with medium confidence

Figure : Test -3 Multiple purchase txt bill

The multiple screenshots above clearly show that a text file is generated with the necessary information when multiple laptops are purchased together. A summary of the purchase is also printed after the purchase is deemed to be complete by the user.

## Test -4: File generation of laptop sales with multiple laptops

|  |  |
| --- | --- |
| Test No: | 4 |
| Objective: | To check file generation for Sales of multiple laptops |
| Action: | * Sell multiple laptops * Check Sales summary in shell * Check .txt bill |
| Expected Result: | The information of multiple Sales must be shown correctly with vat and other required info |
| Actual Result: | All the information was correctly shown |
| Conclusion: | The test is successful |

Table : Test -4

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Test -4 Multiple sales

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Test -4 Multiple sales summary

A screenshot of a computer

Description automatically generated

Figure : Test -4 multiple sales bill generation

As with the purchase test, sales of multiple laptop also generates a single bill with the required information.

## Test -5: Show update in quantity when laptop is bought and sold

|  |  |
| --- | --- |
| Test No: | 5 |
| Objective: | To check update in quantity of txt file when laptop is bought and sold |
| Action: | * Check the txt file and sell laptops * Check the txt file again to confirm update in quantity * Check the txt file again and buy laptop * Check the txt file to confirm update in quantity |
| Expected Result: | The quantity must increase or decrease in accordance to the action performed by the user |
| Actual Result: | Quantity was updated successfully |
| Conclusion: | The test is successful |

Table : Test -5

Below a screenshot of the txt file prior to selling laptop is provided.

A screenshot of a computer program

Description automatically generated with low confidence

Figure : Test -5 txt file before sales

Now we will sell 10 units of xps 13 and 100 units of macbook pro m2 as shown below.

A screenshot of a computer

Description automatically generated with medium confidence

Figure : Test -5 Selling laptop

Now when we look at the txt file again, we can see the quantity of the laptops sold decrease.

A picture containing screenshot, text, font, software

Description automatically generated

Figure : Test -5 Stock after sales

The quantity of the first laptop has decreased by 10 and of the second by 100 as sold in the program.

Now with the above screenshot as reference, we will purchase 20 units of Preditor Triton 500 and 50 units of xps 13.

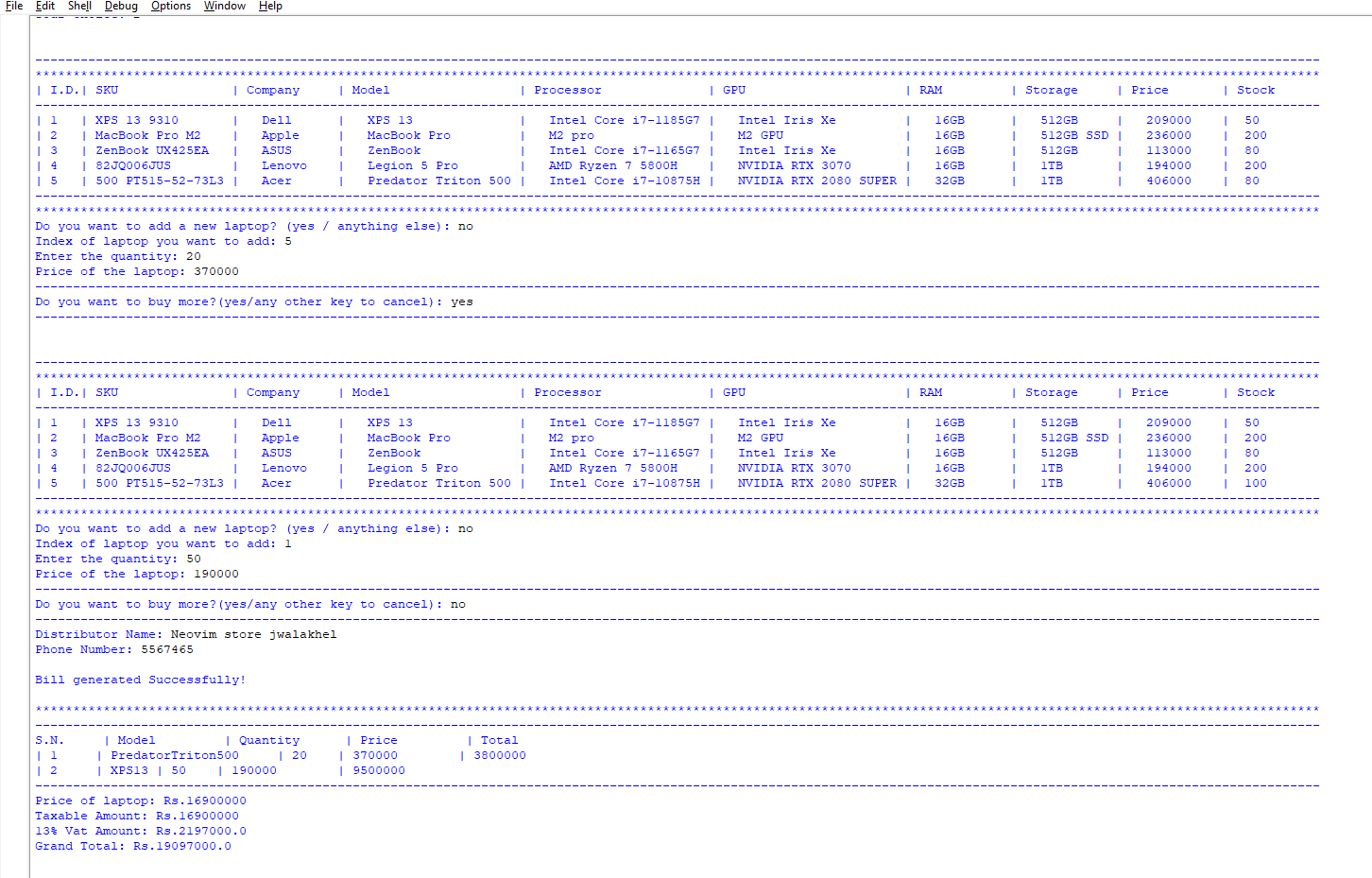


Figure : Test -5 Purchase laptop

Now as we will see below, the quantity of laptops bought has been promptly updated in the txt file.

A picture containing screenshot, software, multimedia software

Description automatically generated

Figure : Test -5 File after purchase

# Conclusion

This was the first-time programming in python in such detail for me. I was pleasantly surprised at how easy to understand and write python code is compared to the other languages we have learned. This does not undermine the level of knowledge you will need in python, and the level of problem-solving skill you need to complete this project properly. I got into this project without knowing anything about file handling and functions in python because of the classes I missed due to sickness. But our tutorial teacher, Mr. Bijay Gautam sir explained to us everything we needed to know and provided a dummy sample of a program which was undoubtedly the most important part of me being able to complete this project.

I decided to use the functional approach because we had another assignment in OOP concept, and I wanted to know which approach I preferred. Since then, I have now completed both of my projects and I can clearly say that I would pick functional approach over object-oriented approach solely because of how little boilerplate it takes to write a simple code.

I have always felt that the hardest part of any project, assignment or work for that matter is starting it. It was the same for me, I would overthink the requirements and the process before even writing a single line of code. That made me procrastinate even more and hindered my development. Once I started breaking down the problem and writing the code, I realized most of it was very easy and had already been thoroughly explained to us by our teachers.

The most difficult part for me to get working was updating the stock properly when buying and selling laptops, which I think is the same for many of the students. We had been storing the information in variables before, which would be disposed of when the program was closed. But since we are working with a txt file as a ‘database’ now, even if the program is terminated our data of laptop stock is still there and can be used again later. It took me some time to figure out how exactly I will tackle this problem. Everyone has their own way of dealing with different situations, and I’ve tried to make my solution as easy to understand and efficient as possible.

I now also understand the growing popularity of python among developers and companies alike. I think how little effort it takes to just start writing python code without any boilerplate or barriers and other little intricacies is what makes it so attractive on top of its easy-to-read syntax. It was a very delightful experience and it helped me gain immense knowledge of not only python but also of developing a working program.

I am very grateful to our Fundamentals Of Computing teachers for helping us acquire the required skill and knowledge about python and programming in general, without which we would not have been able to do this project and further develop ourselves as programmers.