A white cover with blue text

Description automatically generated

# Abstract

A close-up of a text

Description automatically generated

# Objective:

A close-up of a document

Description automatically generated

Table of Contents

[Abstract 2](#_Toc142546308)

[Objective: 2](#_Toc142546309)

[Introduction 5](#_Toc142546310)

[Algorithm 6](#_Toc142546311)

[Pseudocode 8](#_Toc142546312)

[read.py 8](#_Toc142546313)

[write.py 8](#_Toc142546314)

[Operations.py 9](#_Toc142546315)

[main.py 11](#_Toc142546316)

[Flowchart 12](#_Toc142546317)

[Data Structures: 13](#_Toc142546318)

[Program: 15](#_Toc142546319)

[Testing 21](#_Toc142546320)

[Conclusion 24](#_Toc142546321)

**Table of Figures:**

[Figure 1 8](#_Toc142204502)

[Figure 2 8](file:///C:\Users\Saharsh\Desktop\Santosh%20sirhw\bivisha%20hw.docx#_Toc142204503)

[Figure 3 9](#_Toc142204504)

[Figure 4 10](#_Toc142204505)

[Figure 5 10](#_Toc142204506)

[Figure 6 11](#_Toc142204507)

[Figure 7 12](#_Toc142204508)

[Figure 8 14](#_Toc142204509)

[Figure 9 16](#_Toc142204510)

[Figure 10 16](#_Toc142204511)

[Figure 11 16](#_Toc142204512)

[Figure 12 17](#_Toc142204513)

[Figure 13 18](#_Toc142204514)

[Figure 14 18](#_Toc142204515)

[Figure 15 19](#_Toc142204516)

[Figure 16 20](#_Toc142204517)

[Figure 17 21](#_Toc142204518)

[Figure 18 22](#_Toc142204519)

[Figure 19 22](#_Toc142204520)

[Figure 20 23](#_Toc142204521)

[Figure 21 23](#_Toc142204522)

[Figure 22 23](#_Toc142204523)

# Introduction

For this project, I created a program for a computer shop that purchases and resells computers and computers from manufacturers to a variety of consumers, including people and businesses. The software may read a text file holding details about available computers, display them, and modify the file depending on the circumstances of the transaction, such as placing an order with a supplier or making a sale to a client. A note or invoice is generated for each transaction that includes the essential information. After each transaction, the application also updates the stock of a certain computer in the main text file. Using Python, I developed an algorithm, pseudocode, flowchart, and program that included operations for character and string processing, input/output, and data storage. The program shows implementation, accuracy, programming style, handling of exceptions, usability of the user interface and application, testing, and a conclusion.

# Algorithm

1. Start
2. Read the text file containing the computer's information and add it to a list of dictionaries.
3. By looping through the list of dictionaries, display every computer that is accessible in the store.
4. Ask the user if they want to add a new computer, sell a computer to a customer, or order a computer from a manufacturer.
5. If the user wants to order a computer from a manufacturer, then:
   1. Ask the user for the name of the distributor (company).
   2. Inquire about the computer the user wants to order.
   3. Inquire about the brand name from the user.
   4. Request the user's purchase date and time.
   5. Request the user's net amount, which is the entire amount excluding VAT.
   6. Subtract 13% of the net amount from the VAT amount.
   7. Add the net amount and the VAT amount to arrive at the gross amount.
   8. Create a note or invoice with the distributor's name, the computer's brand name, the date and time of the purchase, the net amount, the VAT amount, and the gross amount. Put this in a text file (.txt).
   9. Adding the quantity requested will update the text file containing the computer information's quantity of the ordered computer that is currently available.
6. If the user wants to sell a computer to a customer, then:
   1. Inquire about the brand of the computer the user wants to sell.
   2. Iterate through the list of dictionaries to see if the computer is in stock at the store.
   3. If the computer is available, inquire about the customer's name from the user.
   4. Request the user's purchase date and time.
   5. Multiply the cost of the computer by the number of units sold to arrive at the final sum, excluding shipping costs.
   6. Request the user's shipping costs.
   7. Determine the entire cost of the computers, including the shipping charges.
   8. Create a note or invoice that includes the customer's name, the brand of the computer, the date and time of the transaction, the total money paid (including shipping costs) and the name of the computer. Put this in a text file (.txt).
   9. Subtract the quantity sold from the quantity available of the sold computer in the text file containing information about the computer.
7. If the user wants to add a computer, then:
   1. Ask for the name of the computer.
   2. Ask for the brand of the computer.
   3. Ask for the price of the computer.
   4. Ask for the computer quantity available.
   5. Ask for the computer processor.
   6. Ask for the graphics card in the computer.
8. Ask the user if they want to continue or exit the program.
9. If the user wants to continue, go to step 3.
10. If the user wants to exit, end the program.
11. Stop

# A black text on a white background Description automatically generated

I have divided my program in 4 parts. They are:

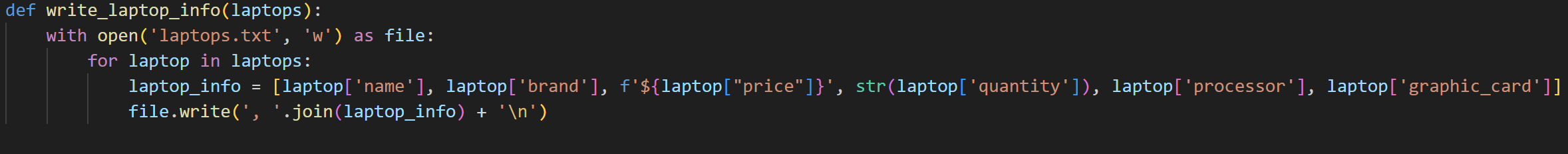
## A black text on a white background Description automatically generated

A screen shot of a computer code

Description automatically generated

Figure

## A black text on a white background Description automatically generated



Figure

## A black text on a white background Description automatically generated

A screen shot of a computer code

Description automatically generated

Figure



Figure

A computer screen shot of a black screen

Description automatically generated

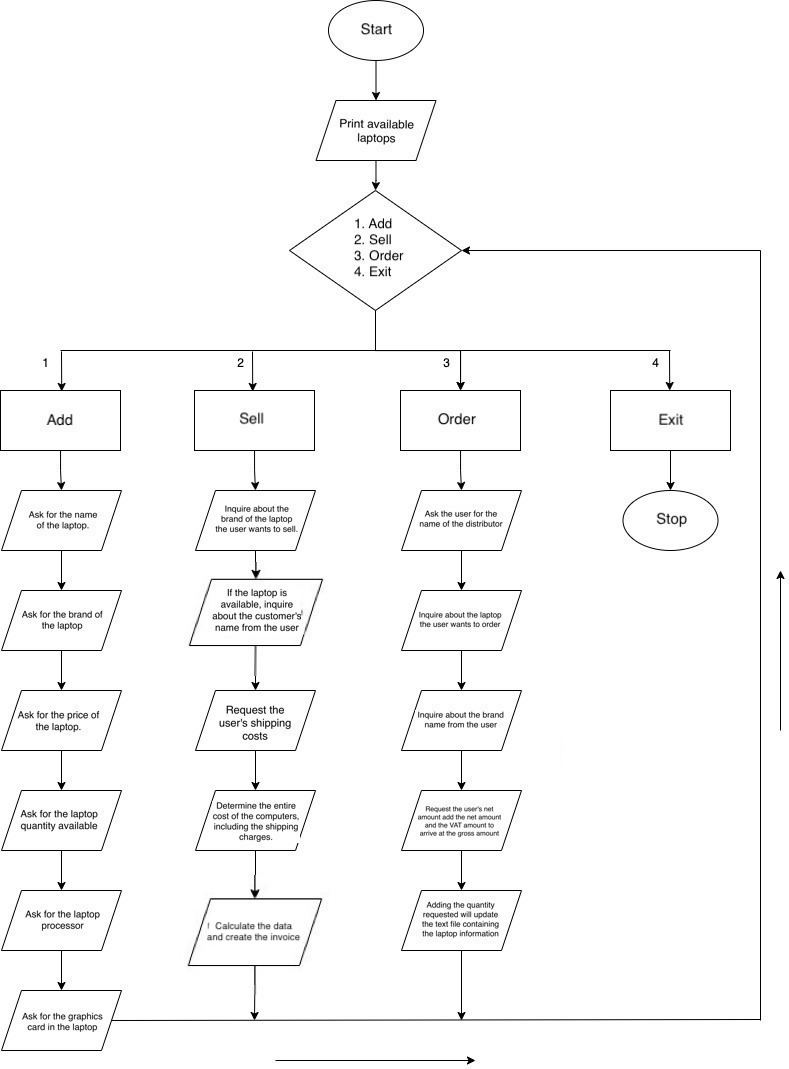
Figure

## 



Figure

# Flowchart



Figure

Data Structures:

1. List: The computers variable is a list of dictionaries. Each dictionary represents a computer and its details such as name, brand, price, quantity, processor, and graphic card.
2. Dictionary: Each computer in the computers list is represented as a dictionary with keys such as 'name', 'brand', 'price', 'quantity', 'processor', and 'graphic\_card'. The dictionary allows the program to store different types of information for each computer and retrieve them easily.
3. String: Several variables in the code are strings, including file which holds the name of the file being read or written to, customer\_name which holds the name of the customer, and transaction\_type which specifies whether the transaction is a sale or an order.
4. Float: The price of each computer in the computers list is stored as a float value. For example, price = 999.99.
5. Integer: The quantity of each computer in the computers list is stored as an integer value. For example, quantity = 10.
6. Datetime: The datetime module is used to generate the current date and time when an invoice is generated. For example, date\_time = datetime.datetime.now().
7. File: The open() function is used to read and write data from and to a file. For example, file = open('computers.txt', 'r'). The 'r' parameter specifies that the file is being opened for reading.

Overall, the code uses a combination of different Python data structures to read, write, and manipulate data related to computers, customers, and transactions.

A picture containing text, diagram, sketch, plan

Description automatically generated

Figure

# Program:

The pseudocode provided above outlines the functions and their use in the program as follows:

1. **main()** function: This feature is activated when the software launches, and it presents the user with a menu of available options. The function invokes further functions to carry out the appropriate action based on the user's selection.
2. **sell\_laptop()** functionWhen the user chooses to sell a laptop, this function is invoked. It requests the user's computer's serial number and checks the inventory to see if it is present. If the computer is available, one is subtracted from the total and the inventory file is updated.
3. **order\_laptop()** function: When a user chooses to order a computer, this function is activated. It requests information about the computer from the user before adding it to the inventory file.
4. **add\_laptop()** function: When the user chooses to add a new computer to the inventory, this function is invoked. It requests information about the computer from the user before adding it to the inventory file.
5. **generate\_invoice()** function: When the user chooses to generate an invoice, this function is invoked. The transaction details, including the total cost, are displayed after reading the inventory file.
6. The main() function is repeatedly invoked by the software to show the menu and request a selection from the user. Which function is invoked to carry out the intended action depends on the user's selection. The application reads and publishes data to the inventory database's text file.

A screen shot of a computer code

Description automatically generated

Figure

The code defines a function read\_laptop\_info() that reads laptop information from a file called "laptops.txt" and returns a list of computer dictionaries, where each dictionary contains information about a single computer including its name, brand, price, quantity, processor, and graphic card.

A black screen with colorful text

Description automatically generated

Figure

The code opens the laptops.txt file and then prints all the information about a single computer including its name, brand, price, quantity, processor, and graphic card.

A black screen with colorful text

Description automatically generated

Figure

The function write\_laptop\_info takes a list of dictionaries containing computer information as input. It opens a file called "laptops.txt" in write mode, and loops through each dictionary in the input list. For each dictionary, it extracts the relevant values and formats them into a list. The function then writes each list to the file in a comma-separated format with a newline character at the end of each line. The end result is a file containing the computer information in a structured format.

A screen shot of a computer program

Description automatically generated

Figure

The function "generate\_invoice" creates a text file invoice containing transaction information including the customer and computer details, date and time, transaction type and amount. The shipping cost and total amount are included in case of a sale, and the VAT and gross amount in case of an order. The invoice file is named uniquely using the customer name, computer name, and transaction time.

A screen shot of a computer code

Description automatically generated

Figure

This code defines a function called "sell\_laptop" that searches for a computer with a given name in a list of computers, and if found and in stock, it sells the computer to a customer by generating an invoice using the "generate\_invoice" function and writing the updated computer information to a file using the "write\_laptop\_info" function. The function takes in the parameters computers, laptop\_name, customer\_name, and an optional parameter, shipping\_cost. The function only outputs print statements that indicate whether a computer was sold or not and does not return any values.

A computer code on a black background

Description automatically generated

Figure

The code defines a function named "order\_laptop" that takes in the parameters computers, laptop\_name, and distributor\_name. The function searches for the computer with the given laptop\_name in the list of computers. If the computer is found, the function increases its quantity by one, generates an invoice using the "generate\_invoice" function with the transaction type set to "Order", and writes the updated computer information to file using the "write\_laptop\_info" function. If the computer is not found, the function prints a message indicating so. The function returns nothing, and the only output is a print statement indicating that a computer was ordered from the distributor.

A screen shot of a computer program

Description automatically generated

Figure

The function "add\_laptop" takes in a list of computers as a parameter and prompts the user to enter various attributes of a new computer. It then creates a new dictionary with the entered attributes and appends it to the list of computers. The function also writes the updated computer information to file and prints a message indicating that the computer has been added to the inventory. This function can be used to add new computers to the list of computers and update the computer information in the file.

A computer screen shot of a program code

Description automatically generated

Figure

This code is the main loop of a computer store program that displays a menu of options to the user, including selling a computer, ordering a computer, adding a computer, or quitting the program. Depending on the user's choice, different functions are called to perform the corresponding action. The program stores the computer information in a file and generates invoices for transactions.

# Testing

A screenshot of a computer program

Description automatically generated with medium confidence

Figure

A screen shot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A screen shot of a computer

Description automatically generated with medium confidence

Figure

A screenshot of a computer

Description automatically generated with medium confidence

Figure

A picture containing text, font, screenshot

Description automatically generated

Figure

# Conclusion

A user-friendly inventory control system for a computer store was created for this assignment. The software enables the purchase, sale, addition of computers, and creation of invoices. Data validity is ensured by error management. By enabling effective inventory management and boosting efficiency, this technology lessens overstocking and stockouts. An organized structure leads to better customer service.

Future improvements might include data-driven reports for better decision-making and a search feature for quick computer research. The task illustrates how technology can be used to streamline corporate operations.