



Module Code & Module Title:

CS4001NT Programming

Assessment Weightage & Type:

60% Individual Coursework

Year and Semester:

2024 Spring

Student Name: Purnima Basnet

London Met ID: 23056706

College ID: np05cp4s240070

Assignment Due Date: 18th August, 2024

Word Count: 7829

Submitted to : AjayRaj Bhattarai

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 mark of zero will be awarded.

Contents

Figure Of Figures	6
Table of Table	8
1. Introduction to Coursework	1
1.1) Python	1
1.2) IDLE (Integrated Development and Learning Environment)	2
1.3) Fundamental of Computing	2
1.4 Introduction to MS-Word	3
1.5) Introduction to Draw.io	4
1.6) Goals	4
1.7) Objectives	5
2.)Introduction to Algorithm	5
2.1) Algorithm	5
3.) Introduction to Pseudocode	8
3.1) Pseudocode of main.py	9
3.2) Pseudocode of read.py	12
3.3) Pseudocode of write.py	12
3.4) Pseudocode of operation.py	13
4) Flowchart	16
5) Data Structure	18
5.1) Linear Data Structure	18
5.1.1) Stack	19
5.1.2) Queues	19
5.1.3) Array	20

5.1.4) Linked List.....	21
5.2) Non-Linear Data Structure.....	22
5.2.1) Trees.....	22
5.2.2) Graph.....	22
5.2.3) Tries.....	22
6) Program	23
6.1) Implementation of the program.....	23
6.1.1) Main.py	23
6.1.3) Display Inventory	24
6.1.4)Operation.py	24
6.1.5) Def Order_from_Manufacturer.....	25
6.1.6) def sell_to_customer.....	25
6.1.7)Read.py.....	26
6.1.8) Write.py.....	26
6.2) User Interface/program usability.....	27
7)Testing	28
7.1) Test 1	28
7.1.1) Showing the list of the Furniture.....	28
7.2) Test 2	29
7.2.1) Selling the furniture to the customer.	29
7.3) Test 3	30
7.3.1) Selling the multiple furniture to the Customer	30
7.4) Test 4	32
7.4.1) Purchasing furniture from the manufacturer.....	32
7.5) Test 5	34

7.5.1) Purchasing furniture from the manufacturer with multiple Purchasing.	34
7.6) Test 6	35
7.6.1) Exception handling (Trying to put the String rather than a integer)	35
7.7) Test7	37
7.7.1) Providing the existed value as a input while selling the products.	37
7.8) Test 8	38
7.8.1) Providing the existed value as a input while Purchasing the products.	38
7.9) Test 9	39
7.9.1) Providing negative Id number while Purchasing	39
7.10) Test 10	40
7.10.1) Providing negative Id number while Selling.....	40
8) Conclusion	41
9) Appendix	42
9.1) Main.py	42
9.2) Operation.py	45
9.3) Read.py	50
9.4) Write.py	51
10) Reference	52
References.....	52
11 Report	55
Student passage FLAGGED	56
Top web match	56
Student passage CITED	56
Top web match	56
Student passage FLAGGED	56

Top web match	57
Student passage FLAGGED	57
Top web match	57
Student passage FLAGGED	57
Top web match	57
Student passage QUOTED	58
Top web match	58
Student passage FLAGGED	58
Top web match	58
Student passage FLAGGED	58
Top web match	58
Student passage FLAGGED	59
Top web match	59
Student passage CITED	59
Top web match	59
Student passage CITED	59
Top web match	60
Student passage FLAGGED	60
Top web match	60
Student passage FLAGGED	60
Top web match	60
Student passage CITED	61
Top web match	61

Figure Of Figures

Figure 1: figure of Python	2
Figure 2: Figure of Ms word	3
Figure 3:Figure of Draw.io	4
Figure 4: Flowchart.....	17
Figure 5:Data Structures	18
Figure 6: Stack Data Structure.	19
Figure 7:Queue Data Structure.	20
Figure 8: Array Data Structure	20
Figure 9: LinkedList Data Structure.....	21
Figure 10:Figure of main.py1	23
Figure 11:Figure of main.py2	24
Figure 12:Display Inventory.....	24
Figure 13: def purchase to customer.....	25
Figure 14: def sales to customer	26
Figure 15: Read.py.....	26
Figure 16:Write.py	27
Figure 17: User Interface.....	27
Figure 18:Selling furniture to Customer.....	29
Figure 19:First Display without decreasing.....	29
Figure 20:After decreasing	30
Figure 21: Selling single items with receipt.	30
Figure 22: first display	31
Figure 23:Display after decreasing.....	31
Figure 24: Selling multiple products	32
Figure 25:Display before purchasing.....	32
Figure 26:Display after purchasing.....	32
Figure 27:Purchasing the Furniture.....	33
Figure 28:Display of purchasing products	34
Figure 29:Exception Handling	35
Figure 30:Display of Exception Handling.....	36

Figure 31:Exception Handling	36
Figure 32:Display of Existed value	37
Figure 33:Providing existed value.	38
Figure 34:Display of providing negative int while purchasing.	39
Figure 35:Display negative value while selling	40

Table of Table

Table 1: List of Furniture.	28
Table 2:Selling the furniture to the customer.	29
Table 3:Selling the multiple products.....	31
Table 4:Purchasing the Furniture.	32
Table 5:Purchasing multiple products.	34
Table 6:Exception Handling	35
Table 7:Providing the existed value.	37
Table 8:Providing existed value while purchasing.....	38
Table 9:Providing negative Id while purchasing.	39
Table 10:Providing negative Id while selling.....	40

1. Introduction to Coursework

This is our 1st coursework of the Fundamentals of Computing (FOC). So, This coursework consist of 60% of total grades. Here, we have got the project of the Python programming languages. We will be doing our code in the IDLE. So, I am glad that we 've receive the coursework of the individual by using the python languages. During, this coursework I learned a lot about the new things of python languages. During this duration I faced a lot of problems and errors and I handle it without any worries and also I took help from the module teachers, instructors & friends. So while doing this I learned to use many tools like the VS Code, Ms Word, idle etc.

So, this coursework is all about creating the BRJ Furniture Store using the python languages. Here, we will create the 4 different folders like: i)Main.py ii) Operation.py iii)Write.py iv) Read.py . Here we will be handling the different files like buying or selling the multiple products. After buying or selling the products the items will be updated in the inventory of furniture. After buying or selling the products the invoice will be generated in the new files with their respective purpose of receipt like purchase or sales invoice. I learned how to handle the error using try except. Here, I have also used the loop for proper functioning.

1.1) Python

Python is a general-purpose, interpreted, high-level programming language. Guido van Rossum was the creator, and it was originally published in 1991.



Figure 1: figure of Python

Python is a great language for both novice and seasoned developers since it places a strong emphasis on code readability and simplicity. Python's copious documentation and simple syntax make it easy to get started with and offer a wealth of resources for learning and development. Python defines code chunks, including loops and functions, using indentation. Take four spaces when indenting text. Syntax mistakes can result from improper indentation. Python is a programming language that facilitates faster work and more effective system integration. (Geeksforgeeks, 2024).

1.2) IDLE (Integrated Development and Learning Environment)

Integrated Development and Learning Environment is what IDLE stands for. The origin of the moniker IDLE is comparable to that of Python.

While the name IDLE was chosen as a tribute to Eric Idle, one of the founding members of the British comic group Monty Python, Guido Van Rossum called Python after the group. Since the 01.5.2b1 release, the standard Python implementation has been included with IDLE. In many Linux, Windows, and Mac distributions, it comes included as an optional element of the Python package.

As demonstrated above, IDLE is an IDE that is both extremely advanced and beginner-friendly. Due to its ease of use, it is strongly suggested for educational. (DataCamp, 2020)

1.3) Fundamental of Computing

The fundamental ideas and concepts that underpin computer science and information technology are referred to as the Fundamentals of Computing (FOC). Understanding

data representation, algorithms, hardware and software components, programming, networking, databases, and theoretical underpinnings are all part of these essentials.

Understanding hardware entails understanding the internal workings of a computer, including the CPU, memory, storage, and input/output devices, as well as how these parts interact with one another. Operating systems, application software, and development tools are all included under the umbrella of software, which emphasizes their functions in managing hardware resources and presenting user interface creation.

When storing and manipulating data in binary form, ASCII/Unicode character encoding ideas are used. This is known as data representation. Data structures and algorithms both use common techniques. (Testbook, 2023)

1.4 Introduction to MS-Word

One word processing program included in the Microsoft Office Suite is Microsoft Word. The Microsoft Office Suite is a collection of robust programs that are intended to give small and large businesses and organizations a wide range of tools for practically any kind of business work. The interface of every program in the Microsoft Office Suite has been designed to be uniform. This obviously facilitates users' ability to switch between applications to complete their varied business activities.



Figure 2: Figure of Ms word

Along with learning the fundamentals of Word, you will also pick up more sophisticated capabilities that will improve your ability to write and create documents, reports, articles,

newsletters, flyers, and letters. You will also pick up a lot of Word shortcuts and features that will help you be more productive. (MyEducator, 2024)

1.5) Introduction to Draw.io

Draw.io is a proprietary charting and diagramming software designed by Seibert Media. You can design a custom layout or use the software's automatic layout tool. They offer hundreds of visual elements and a wide variety of shapes to create a unique diagram or chart. The drag-and-drop functionality facilitates the creation of visually appealing charts and diagrams.



Figure 3: Figure of Draw.io

Depending on your needs, Draw.io offers choices for saving saved charts on a server, in the cloud, or on network storage at a data center. The application comes with a variety of templates, an easy-to-use drag-and-drop interface, and the capacity to interact with other services like GitHub, OneDrive, Google Drive, and more. It can be downloaded as a desktop program or as an online application.

(Computer hope, 2024)

1.6) Goals

This project aims to establish a user-friendly and efficient environment for handling transactions and inventory management. Accurate inventory management is crucial, and this system will automatically adjust stock levels to reflect real-time changes whenever goods are sold or restocked. Our goal is to provide an outstanding user experience that makes it simple for employees to efficiently handle inventory and handle sales. Our primary goal is to satisfy customers by providing them with the greatest items at a price they can afford.

1.7) Objectives

- i) Showing the present furniture items from a furniture.text file, along with ID, product, name, manufacturer, quantity and price.
- ii) When an item is sold or ordered, the text furniture levels of availability are updated to reflect the updated items.
- iii) Creating a entire details invoice for each and every transaction like furniture Id , Manufacturer Name, Price, Quantity, Product Name, Employee Details and total cost should be included to place the order from manufacturer.
- iv) Entire the details of the customer with furniture Id , brand , product name, quantity ,price and total cost before and after shipping cost in which VAT should also be included.
- v) When we buy or sell Multiple products it should be handled in single transaction with its total cost.
- vi) Checking that items are updated in the text furniture file after buying or selling the products.

2.)Introduction to Algorithm

2.1) Algorithm

All branches of information technology heavily rely on algorithms. An algorithm is typically used in mathematics, computer science, and computer programming to describe a brief process that resolves a recurring issue. Algorithms are essential to

automated systems because they provide as guidelines for processing data.

Sorting numerical data sets or more challenging jobs like suggesting user material on social media could both benefit from the application of algorithms. Usually, an algorithm begins with some beginning data and instructions that outline a particular computation. An output is produced by the procedure once the computation is completed.
(techTarget, 1999)

Based on the reference algorithm you provided for managing land rentals and returns, here's an adapted algorithm for the BRJ Furniture Store Management System that reflects the structure and logic:

Step 1: Start

Step 2: Open the Inventory File

Load the inventory file containing all furniture items using ``read_inventory()``.

Step 3: Display Available Furniture

Call ``display_inventory(inventory)`` to display all available furniture items in the store.

Step 4: Ask the User for Their Desired Operation

Prompt the user to choose one of the following options:

1. Order Furniture from Manufacturer
2. Sell Furniture to Customer
3. Exit

If the user chooses Option 1 (Order), go to Step 5

If the user chooses Option 2 (Sell), go to Step 10.

If the user chooses Option 3 (Exit), go to Step 16.

Step 5: Order Furniture

Prompt the employee for their name.

Ask the employee to choose the furniture items they want to order:

Step 6: Ask for the ID of the furniture to be ordered.

Step 7: Ask for the quantity of furniture to be ordered.

Step 8: Check if the selected furniture item exists in the inventory:

If Yes: Proceed to Step 9.

If No: Display an error message and loop back to Step 6.

Step 9: Generate an Order Invoice

Use the ``order_from_manufacturer(details, items)`` function to:

Update the inventory with the ordered quantities.

Generate an invoice file with details such as the employee's name, furniture ID, quantity, manufacturer, total amount, VAT, and shipping cost.

Save the invoice as a ``.txt`` file.

Return to Step 3.

Step 10: Sell Furniture

Prompt the customer for their name.

Ask the customer to choose the furniture items they want to purchase:

Step 11: Ask for the ID of the furniture to be sold.

Step 12: Ask for the quantity of furniture to be sold.

Step 13: Check if the selected furniture item exists in the inventory and has sufficient stock:

If Yes: Proceed to Step 14.

If No: Display an error message and loop back to Step 11.

Step 14: Generate a Sales Invoice

Use the ``sell_to_customer(details, items)`` function to:

Update the inventory by deducting the sold quantities.

Generate an invoice file with details such as the customer's name, furniture ID, quantity, manufacturer, total amount, VAT, and shipping cost.

Save the invoice as a `` .txt`` file.

Return to Step 3.

Step 15: Update Inventory

Update the inventory status in the file using ``write_inventory(inventory)`` after every order or sale transaction.

Step 16: Close the System

If the user chooses to exit, display a goodbye message.

Step 17: End

3.) Introduction to Pseudocode

Pseudocode is an informal method of describing programming that doesn't depend on underlying technological factors or precise programming language syntax. It is employed in the preparation of program rough drafts and/or outlines. Pseudocode omits important elements while summarizing the operation of a program. To make sure that programmers comprehend the requirements of a software project and align code appropriately, system designers create pseudocode.

Program functionality and description are first obtained, then statements are then created using pseudocode to provide the desired outcomes for a program.

Programmers or the designer's team examine and confirm that the detailed pseudocode complies with design requirements. Development can benefit from faults or incorrect program flow detection at the pseudocode stage because it is less expensive to do so than to discover them later. (Economic Times, 2024)

3.1) Pseudocode of main.py

PRINT "Current Inventory:"

PRINT TABLE HEADERS

FOR EACH ITEM IN inventory

PRINT ITEM DETAILS

FUNCTION main()

WHILE True

TRY

PRINT MENU OPTIONS

READ user choice

IF choice IS '1'

CALL read_inventory() AND STORE IN inventory

CALL display_inventory(inventory)

ELSE IF choice IS '2'

INPUT employee_name

INITIALIZE empty list items

WHILE True

TRY

```
    INPUT item ID AND quantity

    APPEND item to items list

    ASK if more items to order

    IF NO

        BREAK loop

    EXCEPT ValueError

        PRINT "Invalid input. Enter numeric values for ID and quantity."

    CALL order_from_manufacturer(employee_name, items)

    PRINT "Order invoice generated."

ELSE IF choice IS '3'

    INPUT customer_name

    INITIALIZE empty list items

    WHILE True

        TRY

            INPUT item ID AND quantity

            APPEND item to items list

            ASK if more items to sell

            IF NO

                BREAK loop

            EXCEPT ValueError

                PRINT "Invalid input. Enter numeric values for ID and quantity."

        CALL sell_to_customer(customer_name, items)

        PRINT "Sale invoice generated."
```

ELSE IF choice IS '4'

PRINT "Thank you for using BRJ Furniture Store Management System.
Goodbye!"

BREAK loop

ELSE

PRINT "Invalid choice. Please try again."

EXCEPT ValueError

PRINT "Invalid input. Enter a valid choice."

IF _name_ == "_main_"

CALL main()

3.2) Pseudocode of read.py

FUNCTION read_inventory(filename='furniture.txt')

INITIALIZE EMPTY DICTIONARY inventory

OPEN FILE with filename IN READ MODE

FOR EACH LINE IN FILE

SPLIT LINE INTO PARTS

PARSE PARTS INTO ID, MANUFACTURER, PRODUCT_NAME, QUANTITY,
AND PRICE

ADD ITEM TO inventory DICTIONARY

RETURN inventory

3.3) Pseudocode of write.py

FUNCTION write_inventory(inventory, filename='furniture.txt')

OPEN FILE with filename IN WRITE MODE

FOR EACH ITEM IN inventory

FORMAT ITEM DETAILS INTO LINE

WRITE LINE TO FILE

FUNCTION update_file(filename, data)

OPEN FILE with filename IN WRITE MODE

FOR EACH LINE IN data

WRITE LINE TO FILE

CLOSE FILE

3.4) Pseudocode of operation.py

IMPORT datetime AND other necessary modules

FUNCTION generate_invoice(transaction_type, details, items, inventory, shipping_cost)

TRY

GET CURRENT datetime

CREATE FILENAME WITH TIMESTAMP

OPEN FILE FOR WRITING

WRITE INVOICE DETAILS

INITIALIZE total_amount TO 0

FOR EACH ITEM IN items

CHECK IF ITEM ID EXISTS IN inventory

CALCULATE amount AND UPDATE total_amount

WRITE ITEM DETAILS TO FILE

CALCULATE VAT

CALCULATE total_amount_with_vat

CALCULATE total_amount_with_shipping

WRITE FINAL AMOUNTS TO FILE

RETURN FILENAME

EXCEPT Exception

PRINT ERROR MESSAGE

RETURN NONE

FUNCTION get_shipping_cost()

WHILE True

TRY

INPUT SHIPPING COST FROM USER

IF SHIPPING COST IS VALID

RETURN SHIPPING COST

ELSE

PRINT "Shipping cost must be greater than 0."

EXCEPT ValueError

PRINT "Invalid input. Enter a valid shipping cost."

FUNCTION order_from_manufacturer(details, items)

TRY

CALL read_inventory() AND STORE IN inventory

CALL get_shipping_cost() AND STORE IN shipping_cost

FOR EACH ITEM IN items

CHECK IF ITEM ID EXISTS IN inventory

UPDATE inventory QUANTITY

CALL write_inventory(inventory)

CALL generate_invoice('order', details, items, inventory, shipping_cost)

EXCEPT Exception

PRINT ERROR MESSAGE

RETURN NONE

FUNCTION sell_to_customer(details, items)

TRY

CALL read_inventory() AND STORE IN inventory

CALL get_shipping_cost() AND STORE IN shipping_cost

FOR EACH ITEM IN items

CHECK IF ITEM ID EXISTS IN inventory

CHECK STOCK LEVEL

UPDATE inventory QUANTITY

CALL write_inventory(inventory)

CALL generate_invoice('sale', details, items, inventory, shipping_cost)

EXCEPT Exception

PRINT ERROR MESSAGE

RETURN NONE

4) Flowchart

A flowchart is a graphical depiction of an algorithm. It is frequently used by programmers to overcome problems through program planning. It makes use of symbols that are linked together to represent the flow of information and processing. The process of creating a flowchart for an algorithm is known as "flowcharting".

Basic Symbols for Flowchart Designs :

Terminal: The oval symbol represents Start, Stop, and Halt in a program's logic flow. A pause/halt is commonly employed in program logic under certain error situations.

Terminal is the first and last symbols in the flowchart.

Input/Output: A parallelogram represents any function of input/output kind. In a flowchart, parallelograms show program instructions that accept input from input devices and output to output devices.

Processing: Each block represents an arithmetic operation. All arithmetic operations, such as adding, subtracting, multiplying, and dividing, are represented by an action or process symbol. (Geeksforgeeks, 2024)

Text

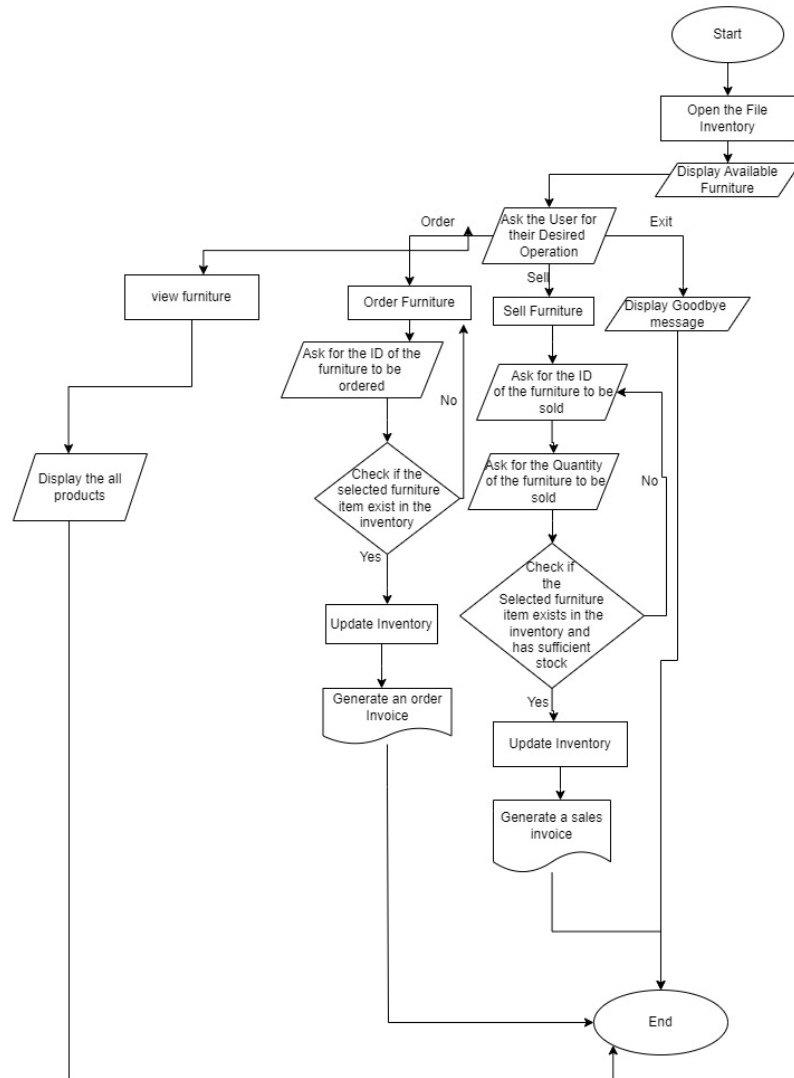


Figure 4: Flowchart

5) Data Structure

A specific format for data organization, processing, retrieval, and storage is called a data structure. Data structures come in a variety of fundamental and sophisticated forms, all intended to organize data for a particular use. Users may easily obtain and operate with the data they require thanks to data structures. Above all, data structures organize information so that both computers and people can comprehend it more easily. A data structure may be chosen or created in computer science and computer programming with the intention of storing data for use with different algorithms; these combinations of data structures and algorithms are known as data structures and algorithms (DSA). In certain instances, the design of the data structure and the algorithm's fundamental operations are closely related. Similarly there are two types of data structure: i) Linear Data Structure and ii) Non- linear data structure(techTarget,2005)

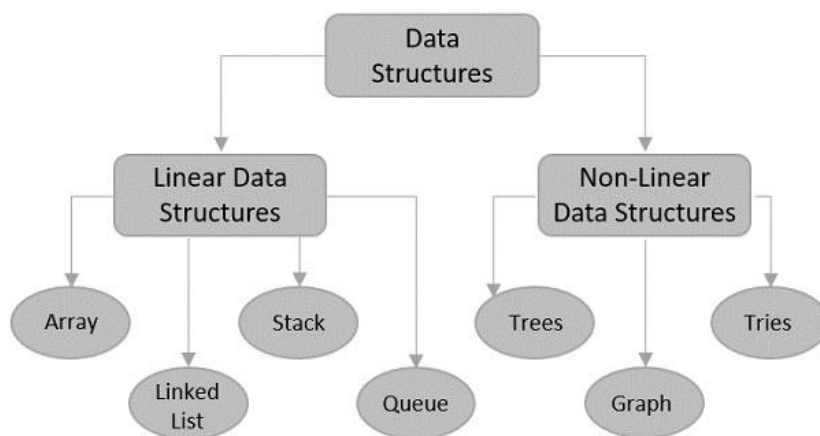


Figure 5:Data Structures

5.1) Linear Data Structure

A data structure that stores information either sequentially or linearly is called a linear data structure. Data is structured in a linear data structure so that each element is next

to its predecessor and so on. In order to traverse all of the data in a single pass, it contains the data at a single level. (UnStop, 2024)

5.1.1) Stack

Data can be pushed or popped from one end of the stack by users. Users can use the push and pop operations to add and remove data from the stack, respectively. The stack adheres to the last in, first out (LIFO) principle. From the top of the stack, users have linear access to all of the stack data. The stack data structure has numerous uses in real-world scenarios. For instance, the stack is used by the All web browser to store backward and forward activities. (UnStop, 2024)

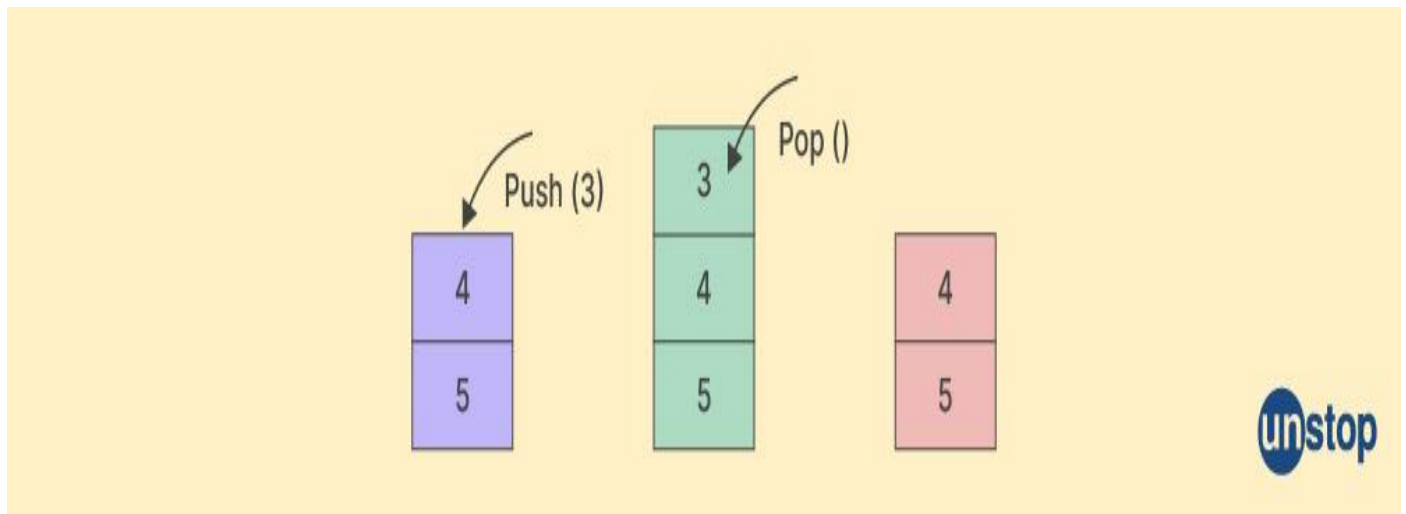
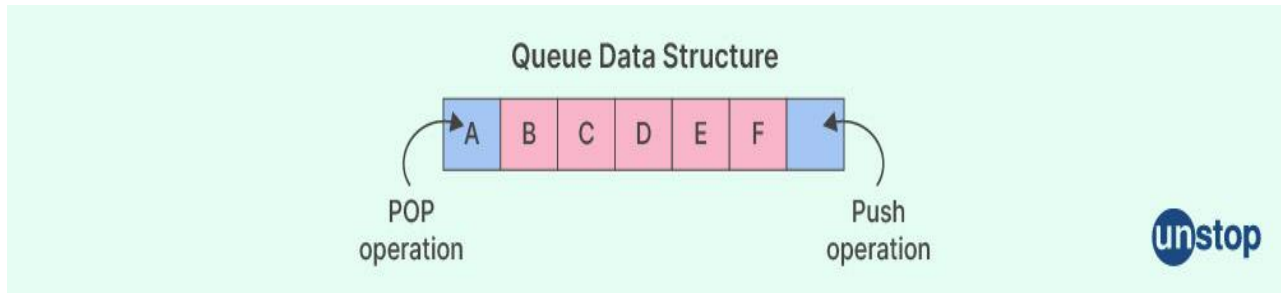


Figure 6: Stack Data Structure.

5.1.2) Queues

The data is kept in a linear sequence in a queue data structure. The queue data structure adheres to the first-in-first-out (FIFO) rule. With two ends, it resembles the stack data structure. We can use the enqueue and deque methods to perform insertion operations from the back of the queue and front-of-queue deletion operations, respectively. One of the best real-world illustrations of a queue is an escalator.

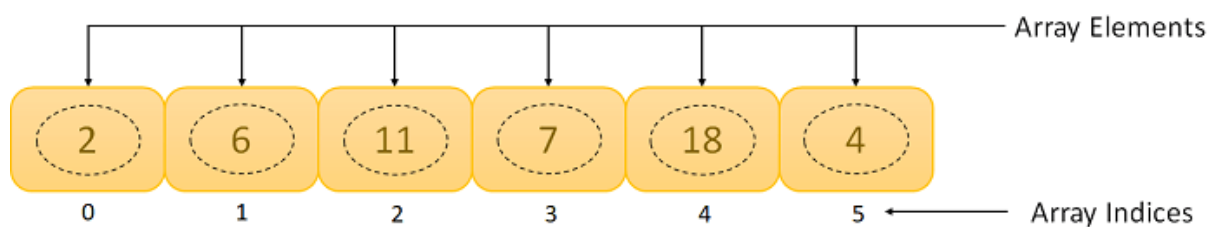
(UnStop, 2024)

*Figure 7: Queue Data Structure.*

5.1.3) Array

Elements in an array are kept in memory at contiguous locations, meaning they are always next to one another. It is homogenous, which means that only the characters, numbers, and other data types previously mentioned are contained within a structure.) Different linear data structures can be implemented in different programming languages and hence be homogeneous or heterogeneous.

Every element in an array has an index, which is a numerical identifier that shows the element's location. For the initial element, indexing starts at 0, and it increases progressively up to the array size minus one. (altexSoft, 2024)

*Figure 8: Array Data Structure*

5.1.4) Linked List

Another example of a linear data structure that is used to dynamically hold a group of data pieces is a linked list. The Nodes in this data structure represent the data items, which are connected by links or pointers. Every node has two fields: a pointer field that contains the address of the nodes that come after it in the list, and an information field that holds the actual data. Since it points to nothing, the pointer of the final node in the linked list is a null pointer. In contrast to arrays, a linked list's size can be dynamically changed by the user to suit specific needs. (javatPoint, 2024)

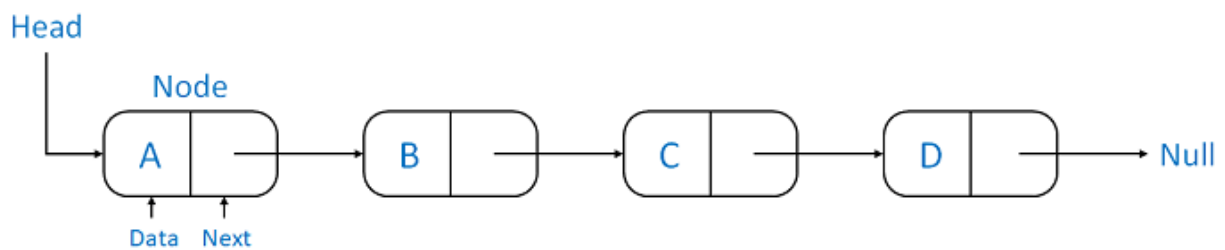


Figure 9: LinkedList Data Structure.

5.2) Non-Linear Data Structure

Data is stored in non-linear data structures as a hierarchy. As a result, the data are difficult to navigate and are available on several levels, unlike linear data structures. They are made to get around the drawbacks and restrictions of linear data structures, nevertheless. For example, memory allocation is the primary drawback of linear data structures. Each element in linear data structures consumes an entire memory block since the data is allocated progressively in these structures. On the other hand, the excess memory in the block is squandered if the data consumes less memory than the allocated block can hold. Non-linear data structures are therefore introduced. They make effective use of the memory and reduce the complexity of space. (Tutorials Point, 2024)

5.2.1) Trees

A tree data structure is made up of several nodes connected to one another. A tree's hierarchical structure resembles the relationship between a parent and kid. Every parent-child node relationship in the tree has a single connection because of the way its structure is constructed. There should be just one path from the tree's root to each node. There are many different kinds of trees, such as AVL trees, binary trees, binary search trees, etc., depending on their structural makeup. (Geeksforgeeks, 2024)

5.2.2) Graph

Graphs are a class of non-linear data structures with a fixed number of edges and vertices. Data is stored by the vertices, also known as the nodes, while the edges display the relationships between the vertices. A graph and a tree differ in that a graph does not have any set rules for how nodes are connected to one another. The graphs can be used to illustrate real-world issues with social networks, phone networks, and other networks. (Geeksforgeeks, 2024)

5.2.3) Tries

Strings are stored in a data structure called a tries, or keyword tree, as data pieces that can be arranged visually in a graph. (techTarget, 2005)

6) Program

6.1) Implementation of the program.

6.1.1) Main.py

An interactive menu system for running a furniture business has been included by this Python code. The application welcomes the user and displays a menu with four important options when it first launches: showcasing the furniture that is available, handling furniture sales, putting new orders, and closing the application. It calls functions from the `furniture`, `sell`, and `order` modules to carry out the associated responsibilities based on the input. The prompt reappear[s] if the user submits incorrect data or makes a format error.

```
def main():
    while True:
        print("Welcome to BRJ Furniture Store Management System")
        print("Please select an option:")
        print("1. Display all available furniture")
        print("2. Order furniture from manufacturer")
        print("3. Sell furniture to customer")
        print("4. Exit")

        choice = input("Enter your choice (1/2/3/4): ")

        if choice == '1':
            inventory = read_inventory()
            display_inventory(inventory)
        elif choice == '2':
            employee_name = input("Enter the name of the employee ordering the furniture: ")
            details = f"Employee: {employee_name}"
            items = []

            while True:
                try:
                    id = int(input("Enter the ID of the furniture to order: "))
                    quantity = int(input("Enter the quantity to order: "))
                    items.append({'id': id, 'quantity': quantity})
                except ValueError:
                    print("Invalid input. Please enter numeric values for ID and quantity.")
                    continue

            more = input("Do you want to order more items? (yes/no): ")
            if more.lower() != 'yes':
                break

            order_invoice = order_from_manufacturer(details, items)
            print(f"Order invoice generated: {order_invoice}")
        elif choice == '3':
            customer_name = input("Enter the name of the customer: ")
            details = f"Customer: {customer_name}"
            items = []

            while True:
                try:
```

Figure 10:Figure of main.py1

```

main.py - C:\Users\Owner\OneDrive\Desktop\FOC CourseWork\main.py (3.12.4)
File Edit Format Run Options Window Help

    more = input("Do you want to order more items? (yes/no): ")
    if more.lower() != 'yes':
        break

    order_invoice = order_from_manufacturer(details, items)
    print(f"Order invoice generated: {order_invoice}")
elif choice == '3':
    customer_name = input("Enter the name of the customer: ")
    details = f"Customer: {customer_name}"
    items = []

    while True:
        try:
            id = int(input("Enter the ID of the furniture to sell: "))
            quantity = int(input("Enter the quantity to sell: "))
            items.append({'id': id, 'quantity': quantity})
        except ValueError:
            print("Invalid input. Please enter numeric values for ID and quantity.")
            continue

    more = input("Do you want to sell more items? (yes/no): ")
    if more.lower() != 'yes':
        break

    sale_invoice = sell_to_customer(details, items)
    print(f"Sale invoice generated: {sale_invoice}")
elif choice == '4':
    print("Thank you for using BRJ Furniture Store Management System. Goodbye!")
    break
else:
    print("Invalid choice. Please try again.")

# Ask if the user wants to choose more options
more_options = input("Do you want to choose from BRJ management store (yes/no): ")
if more_options.lower() != 'yes':
    print("Thank you for using BRJ Furniture Store Management System. Goodbye!")
    break

if __name__ == "__main__":
    main()

```

Figure 11:Figure of main.py2

6.1.3) Display Inventory

This section of code is a part of the display_stock function, which is in charge of presenting an inventory of items of furniture in a well formatted table.

```

def display_inventory(inventory):
    print("\nCurrent Inventory:")
    print(f"{'ID':<5} {'Manufacturer':<25} {'Product':<20} {'Quantity':<10} {'Price':<10}")
    print("-" * 75)
    for id, details in inventory.items():
        print(f"{id:<5} {details['manufacturer']:<25} {details['product_name']:<20} {details['quantity']:<10} ${details['price']:<10.2f}")
    print("\n")

```

Figure 12:Display Inventory.

6.1.4)Operation.py

Here, in this code we have define the sells function and purchase function.The function are defined below:

6.1.5) Def Order_from_Manufacturer

By keeping track of inventory changes and generating invoices, the `purchase_furniture` function streamlines the furniture ordering process. An empty list is formed to keep the information about the requested furniture pieces, and the user is prompted to provide the distributor's name through a utility function. After that, the code goes into a loop that supports numerous ordering. By using loop we can order multiple items from the furniture store according to the user needs. If entered ID matches the item ID then it will update items in the inventory.

```
def order_from_manufacturer(details, items):
    try:
        inventory = read_inventory()
        shipping_cost = get_shipping_cost() # Get the validated shipping cost from the user
        for item in items:
            id = item['id']
            quantity = item['quantity']
            if id in inventory:
                inventory[id]['quantity'] += quantity
            else:
                print(f"Error: Furniture ID {id} does not exist.")
        write_inventory(inventory)
        return generate_invoice('order', details, items, inventory, shipping_cost)
    except Exception as e:
        print(f"An error occurred during the order process: {e}")
        return None
```

Figure 13: def purchase to customer.

6.1.6) def sell_to_customer

By keeping track of inventory changes and generating invoices, the `sell_furniture` function streamlines the furniture selling process. An empty list is formed to keep the information about the requested furniture pieces, and the user is prompted to provide the distributor's name through a utility function. After that, the code goes into a loop that supports numerous selling. By using loop we can order multiple items from the furniture store according to the user needs. If entered ID matches the item ID then it will update items in the inventory.

```

        print(e)
def order_from_manufacturer(details, items):
    try:
        inventory = read_inventory()
        shipping_cost = get_shipping_cost() # Get the validated shipping cost from the user
        for item in items:
            id = item['id']
            quantity = item['quantity']
            if id in inventory:
                inventory[id]['quantity'] += quantity
            else:
                print(f"Error: Furniture ID {id} does not exist.")
        write_inventory(inventory)
        return generate_invoice('order', details, items, inventory, shipping_cost)
    except Exception as e:
        print(f"An error occurred during the order process: {e}")
        return None

```

Figure 14: def sales to customer

6.1.7) Read.py

The read_furniture_data function takes in furniture data from a file and assigns it to a list of dictionaries, each of which corresponds to a specific item of furniture.

```

File Edit Format Run Options Window Help
def read_inventory(filename='furniture.txt'):
    inventory = {}
    with open(filename, 'r') as file:
        for line in file:
            parts = line.strip().split(',')
            id = int(parts[0])
            manufacturer = parts[1]
            product_name = parts[2]
            quantity = int(parts[3])
            price = float(parts[4].strip("$"))

            inventory[id] = {
                'manufacturer': manufacturer,
                'product_name': product_name,
                'quantity': quantity,
                'price': price
            }
    return inventory

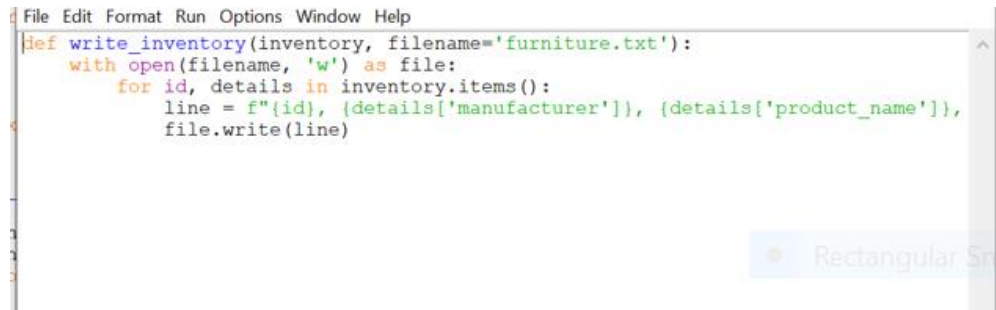
```

Figure 15: Read.py

6.1.8) Write.py

The updated furniture list can be saved to a specified file using the write_to_file function. It opens the file in write mode ('w'), iterates over each piece of furniture, and writes the information (ID, name, manufacturer, quantity, and price) to the

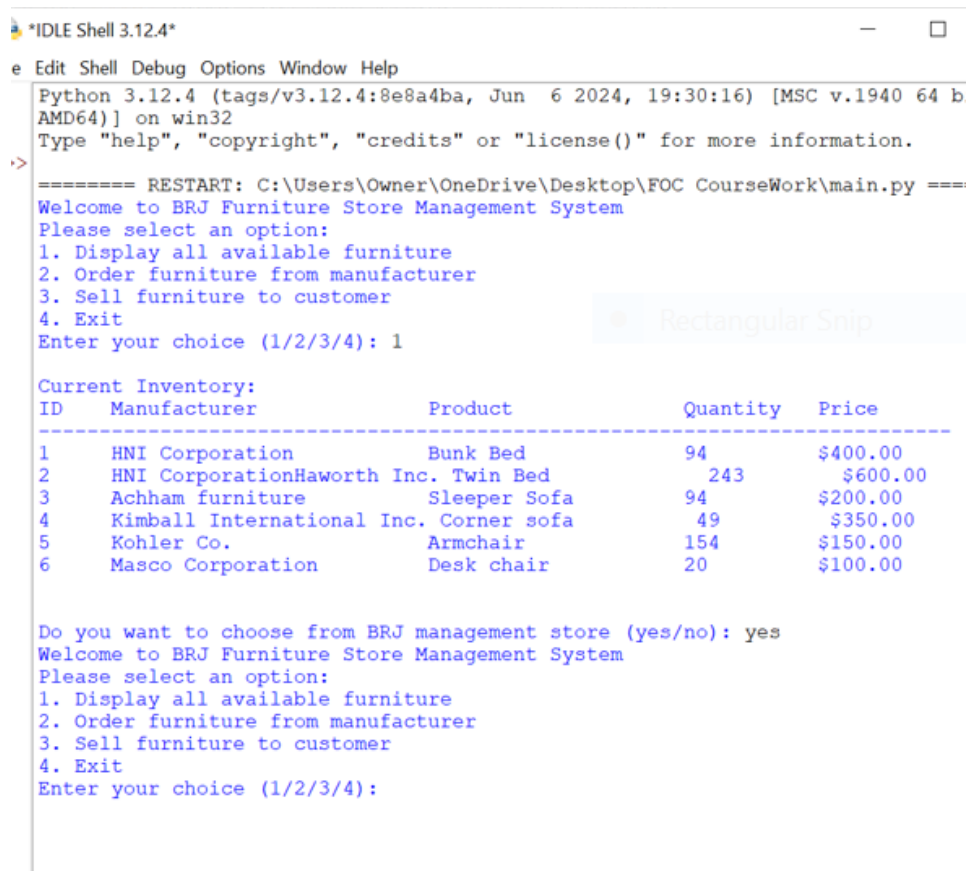
file in a structured string.



```
File Edit Format Run Options Window Help
def write_inventory(inventory, filename='furniture.txt'):
    with open(filename, 'w') as file:
        for id, details in inventory.items():
            line = f"{id}, {details['manufacturer']}, {details['product_name']},
            file.write(line)
```

Figure 16: Write.py

6.2) User Interface/program usability



```
*IDLE Shell 3.12.4*
e Edit Shell Debug Options Window Help
Python 3.12.4 (tags/v3.12.4:8e8a4ba, Jun 6 2024, 19:30:16) [MSC v.1940 64 b
AMD64] on win32
Type "help", "copyright", "credits" or "license()" for more information.
->
===== RESTART: C:\Users\Owner\OneDrive\Desktop\FOC CourseWork\main.py =====
Welcome to BRJ Furniture Store Management System
Please select an option:
1. Display all available furniture
2. Order furniture from manufacturer
3. Sell furniture to customer
4. Exit
Enter your choice (1/2/3/4): 1

Current Inventory:
ID      Manufacturer      Product      Quantity  Price
-----
1       HNI Corporation      Bunk Bed      94        $400.00
2       HNI CorporationHaworth Inc. Twin Bed      243       $600.00
3       Achham furniture      Sleeper Sofa      94        $200.00
4       Kimball International Inc. Corner sofa      49        $350.00
5       Kohler Co.      Armchair      154       $150.00
6       Masco Corporation      Desk chair      20        $100.00

Do you want to choose from BRJ management store (yes/no): yes
Welcome to BRJ Furniture Store Management System
Please select an option:
1. Display all available furniture
2. Order furniture from manufacturer
3. Sell furniture to customer
4. Exit
Enter your choice (1/2/3/4):
```

Figure 17: User Interface

7)Testing

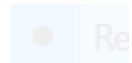
7.1) Test 1

7.1.1) Showing the list of the Furniture

Objective	To display the all list of the Furniture.
Action	A Number Integer is input
Expected result	List of furniture should be displayed.
Actual result	List of furniture be displayed.
Conclusion	The test was successful.

Table 1: List of Furniture.

```
3. Sell furniture to customer
4. Exit
Enter your choice (1/2/3/4): 1
```



Current Inventory:

ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	211	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	64	\$350.00
5	Kohler Co.	Armchair	92	\$150.00
6	Masco Corporation	Desk chair	30	\$100.00

Figure 18: Selling furniture to Customer

7.2) Test 2

7.2.1) Selling the furniture to the customer.

Objective	Selling the furniture to the Customer.
Action	Enter your choice (1/2/3/4): 3 Enter the name of the customer: Purnima Enter the ID of the furniture to sell: 2 Enter the quantity to sell: 4 Do you want to sell more items? (yes/no): no Enter the shipping cost: \$300
Expected Result	The quantity of the selected goods must be decreased.
Actual result	The quantity of the selected goods decreased.
Conclusion	The test was successful.

Table 2: Selling the furniture to the customer.

Current Inventory:				
ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	101	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	212	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	73	\$350.00
5	Kohler Co.	Armchair	32	\$150.00
6	Masco Corporation	Desk chair	40	\$100.00

Figure 19: First Display without decreasing.

Current Inventory:

ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	101	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	208	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	73	\$350.00
5	Kohler Co.	Armchair	32	\$150.00
6	Masco Corporation	Desk chair	40	\$100.00

Figure 20:After decreasing



Figure 21: Selling single items with receipt.

7.3) Test 3

7.3.1) Selling the multiple furniture to the Customer

Objective	Selling the multiple furniture to the Customer
Action	<p>Enter your choice (1/2/3/4): 3</p> <p>Enter the name of the customer: Hari</p> <p>Enter the ID of the furniture to sell: 4</p> <p>Enter the quantity to sell: 15</p> <p>Do you want to sell more items? (yes/no): yes</p> <p>Enter the ID of the furniture to sell: 6</p> <p>Enter the quantity to sell: 10</p>

	Do you want to sell more items? (yes/no): no Enter the shipping cost: \$300
Expected Result	The quantity of the selected goods must be decreased.
Actual Result	The quantity of the selected goods decreased.
Conclusion	The test was successful.

Table 3: Selling the multiple products.

Current Inventory:				
ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	101	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	208	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	73	\$350.00
5	Kohler Co.	Armchair	32	\$150.00
6	Masco Corporation	Desk chair	40	\$100.00

Figure 22: first display

Current Inventory:				
ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	101	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	208	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	58	\$350.00
5	Kohler Co.	Armchair	32	\$150.00
6	Masco Corporation	Desk chair	30	\$100.00

Figure 23: Display after decreasing

Sale Invoice

Date and Time: 2024-08-13 10:53:32

Details: Customer: Hari

Items:

ID: 4, Manufacturer: Kimball International Inc., Product: Corner sofa, Quantity: 15, Price: \$350.00, Amount: \$5250.00

ID: 6, Manufacturer: Masco Corporation, Product: Desk chair, Quantity: 10, Price: \$100.00, Amount: \$1000.00

=====

Total Amount: \$6250.00

VAT (13%): \$812.50

Total Amount with VAT: \$7062.50

Shipping Cost: \$300.00

Total Amount to be Paid: \$7362.50

Figure 24: Selling multiple products

7.4) Test 4**7.4.1) Purchasing furniture from the manufacturer.**

Objective	Purchasing furniture from the manufacturer
Action	Enter your choice (1/2/3/4): 2 Enter the name of the employee ordering the furniture: kkk Enter the ID of the furniture to order: 5 Enter the quantity to order: 15 Do you want to order more items? (yes/no): no Enter the shipping cost: \$600
Expected Result	List of furniture must be presented
Actual Result	List of furniture be presented
Conclusion	The test was successful.

Table 4:Purchasing the Furniture.

Current Inventory:				
ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	208	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	64	\$350.00
5	Kohler Co.	Armchair	32	\$150.00
6	Masco Corporation	Desk chair	30	\$100.00

Figure 25:Display before purchasing

Current Inventory:				
ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	208	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	64	\$350.00
5	Kohler Co.	Armchair	47	\$150.00
6	Masco Corporation	Desk chair	30	\$100.00

Figure 26:Display after purchasing

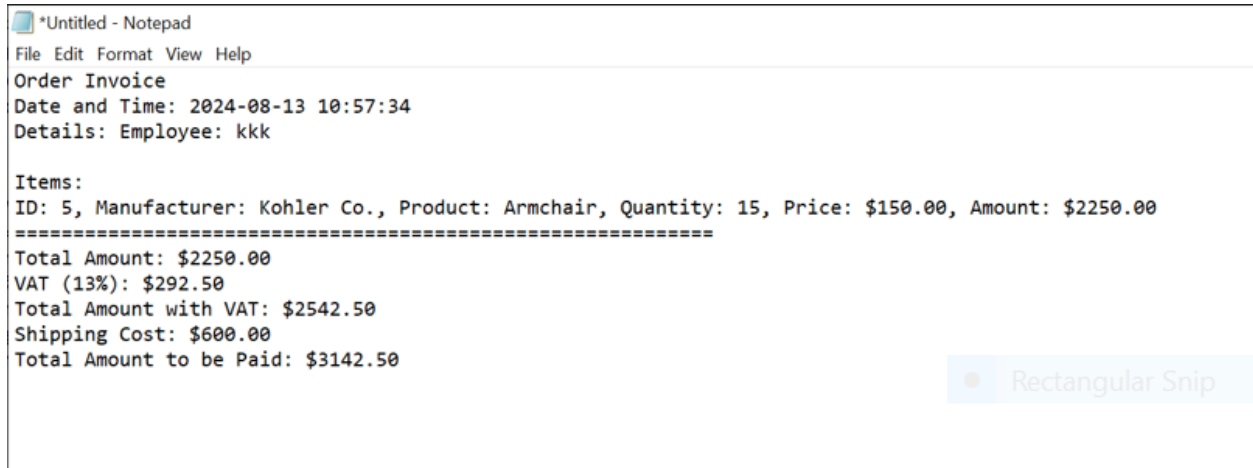


Figure 27: Purchasing the Furniture.

7.5) Test 5

7.5.1) Purchasing furniture from the manufacturer with multiple Purchasing.

Objective	Purchasing furniture from the manufacturer with multiple order.
Action	Enter your choice (1/2/3/4): 2 Enter the name of the employee ordering the furniture: rrr Enter the ID of the furniture to order: 2 Enter the quantity to order: 3 Do you want to order more items? (yes/no): yes Enter the ID of the furniture to order: 5 Enter the quantity to order: 45 Do you want to order more items? (yes/no): no Enter the shipping cost: \$400
Expected Result	The quantity of the multiple order must be increased.
Actual Result	The quantity of the multiple order increased.
Conclusion	The test was Successful.

Table 5:Purchasing multiple products.

Current Inventory:				
ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	208	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	64	\$350.00
5	Kohler Co.	Armchair	47	\$150.00
6	Masco Corporation	Desk chair	30	\$100.00

Figure 28:Display of purchasing products

7.6) Test 6

7.6.1) Exception handling (Trying to put the String rather than a integer) .

Objective	To show an error while using String rather than a number Integer.
Action	A String is provided as input rather than a number Integer .
Expected Result	Error message must be displayed.
Actual Result	Error message was displayed in the terminal.
Conclusion	The test was successful.

Table 6:Exception Handling

```

----- RESUME: C:\Users\owner\OneDrive\Desktop\FCC COURSEWORK\
Welcome to BRJ Furniture Store Management System
Please select an option:
1. Display all available furniture
2. Order furniture from manufacturer
3. Sell furniture to customer
4. Exit
Enter your choice (1/2/3/4): A
Invalid choice. Please try again.
Do you want to choose from BRJ management store (yes/no): |

```

Figure 29:Exception Handling

Current Inventory:

ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	211	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	64	\$350.00
5	Kohler Co.	Armchair	92	\$150.00
6	Masco Corporation	Desk chair	30	\$100.00

Figure 30:Display of Exception Handling

```

*Untitled - Notepad
File Edit Format View Help
5 Order Invoice
Date and Time: 2024-08-13 11:00:27
Details: Employee: rrr

Items:
ID: 2, Manufacturer: HNI CorporationHaworth Inc., Product: Twin Bed, Quantity: 3, Price: $600.00, Amount: $1800.00
ID: 5, Manufacturer: Kohler Co., Product: Armchair, Quantity: 45, Price: $150.00, Amount: $6750.00
=====
Total Amount: $8550.00
VAT (13%): $1111.50
Total Amount with VAT: $9661.50
Shipping Cost: $400.00
Total Amount to be Paid: $10061.50

```

Figure 31:Exception Handling

7.7) Test7

7.7.1) Providing the existed value as a input while selling the products.

Objective	Providing the existed value as a input
Action	<p>Enter your choice (1/2/3/4): 3</p> <p>Enter the name of the customer: hh</p> <p>Enter the ID of the furniture to sell: 17</p> <p>Enter the quantity to sell: 5</p> <p>Do you want to sell more items? (yes/no): no</p> <p>Enter the shipping cost: \$500</p> <p>Error: Furniture ID 17 does not exist.</p> <p>An error occurred while generating the invoice:</p> <p>Error: Furniture ID 17 does not exist in the inventory.</p> <p>Sale invoice generated: None</p> <p>Do you want to choose from BRJ management store (yes/no): no</p> <p>Thank you for using BRJ Furniture Store Management System. Goodbye!</p>
Expected Result	Error should be occurred
Exact Result	Error occurred
Conclusion	Test was successful.

Table 7: Providing the existed value.

Current Inventory:				
ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	210	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	49	\$350.00
5	Kohler Co.	Armchair	152	\$150.00
6	Masco Corporation	Desk chair	20	\$100.00

Figure 32: Display of Existed value

7.8) Test 8

7.8.1) Providing the existed value as a input while Purchasing the products.

Objectives	Providing the existed value as a input
Action	Enter your choice (1/2/3/4): 2 Enter the name of the employee ordering the furniture: fff Enter the ID of the furniture to order: 67 Enter the quantity to order: 5 Do you want to order more items? (yes/no): no Enter the shipping cost: \$200 Error: Furniture ID 67 does not exist. An error occurred while generating the invoice: Error: Furniture ID 67 does not exist in the inventory. Order invoice generated: None
Expected Result	Error Should be Occured
Exact Result	Error Occured
Conclusion	Test was successful.

Table 8: Providing existed value while purchasing

Current Inventory:

ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	210	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	49	\$350.00
5	Kohler Co.	Armchair	152	\$150.00
6	Masco Corporation	Desk chair	20	\$100.00

Figure 33: Providing existed value.

7.9) Test 9

7.9.1) Providing negative Id number while Purchasing

Objective	Negative Id number while Purchasing
Action	<p>Enter your choice (1/2/3/4): 2</p> <p>Enter the name of the employee ordering the furniture: eee</p> <p>Enter the ID of the furniture to order: -7</p> <p>Enter the quantity to order: 4</p> <p>Do you want to order more items? (yes/no): no</p> <p>Enter the shipping cost: \$600</p> <p>Error: Furniture ID -7 does not exist.</p> <p>An error occurred while generating the invoice: Error: Furniture ID -7 does not exist in the inventory.</p> <p>Order invoice generated: None</p> <p>Do you want to choose from BRJ management store (yes/no): no</p> <p>Thank you for using BRJ Furniture Store Management System. Goodbye!</p>
Expected Result	Error should be Occured
Exact Result	Error Occured
Conclusion	Test was succesful

Table 9:Providing negative Id while purchasing.

Current Inventory:

ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	210	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	49	\$350.00
5	Kohler Co.	Armchair	152	\$150.00
6	Masco Corporation	Desk chair	20	\$100.00

Figure 34:Display of providing negative int while purchasing.

7.10) Test 10

7.10.1) Providing negative Id number while Selling

Objective	Providing negative Id number while selling
Action	<p>Enter your choice (1/2/3/4): 3</p> <p>Enter the name of the customer: rrr</p> <p>Enter the ID of the furniture to sell: -9</p> <p>Enter the quantity to sell: 5</p> <p>Do you want to sell more items? (yes/no): no</p> <p>Enter the shipping cost: \$45</p> <p>Error: Furniture ID -9 does not exist.</p> <p>An error occurred while generating the invoice: Error:</p> <p>Furniture ID -9 does not exist in the inventory.</p> <p>Sale invoice generated: None</p> <p>Do you want to choose from BRJ management store (yes/no): no</p> <p>Thank you for using BRJ Furniture Store Management System. Goodbye!</p>
Expected Result	Error should be Occured
Exact Result	Error Occured
Conclusion	Test was successful.

Table 10:Providing negative Id while selling

Current Inventory:

ID	Manufacturer	Product	Quantity	Price
1	HNI Corporation	Bunk Bed	106	\$400.00
2	HNI CorporationHaworth Inc.	Twin Bed	210	\$600.00
3	Achham furniture	Sleeper Sofa	93	\$200.00
4	Kimball International Inc.	Corner sofa	49	\$350.00
5	Kohler Co.	Armchair	152	\$150.00
6	Masco Corporation	Desk chair	20	\$100.00

Figure 35:Display negative value while selling

8) Conclusion

Through this project, I was able to improve my programming skills and have a deeper understanding of Python. One of the most widely used languages is Python, mostly because of its clear and simple syntax. Such coursework, in my opinion, enables students to apply their creativity and experience in real-world contexts. I received the support of my teacher and friends, which helped me stay confident and finish the task effectively even though the project had its share of difficulties and may be confusing at times. By the time the project was over, I was at ease handling Python's modules and methods. I had some trouble at first understanding functions and modules, but now I find them to be really simple to utilize.

Additionally, I improved my understanding of program flow and logic. The project's main goal was to familiar us with Python file management. File management operations including creating, opening, reading, writing, and appending files are made easier by Python's built-in methods. Python handles text and binary files differently, which is a significant line to make. This character marks the beginning of a new line and the end of an existing one. Overall, this project has given me skills and information that will be extremely useful for my future studies, especially since Python has grown to be a language that I genuinely like working with.

9) Appendix

9.1) Main.py

```
from operation import order_from_manufacturer, sell_to_customer

from read import read_inventory


def display_inventory(inventory):

    print("\nCurrent Inventory:")

    print(f"{'ID':<5} {'Manufacturer':<25} {'Product':<20} {'Quantity':<10} {'Price':<10}")

    print("-" * 75)

    for id, details in inventory.items():

        print(f"{'id':<5} {'details['manufacturer']':<25} {'details['product_name']':<20} {"
        {'details['quantity']':<10} ${details['price']:<10.2f}")

    print("\n")


def main():

    while True:

        print("Welcome to BRJ Furniture Store Management System")

        print("Please select an option:")

        print("1. Display all available furniture")

        print("2. Order furniture from manufacturer")

        print("3. Sell furniture to customer")

        print("4. Exit")
```

```
choice = input("Enter your choice (1/2/3/4): ")

if choice == '1':

    inventory = read_inventory()

    display_inventory(inventory)

elif choice == '2':

    employee_name = input("Enter the name of the employee ordering the furniture: ")

    details = f"Employee: {employee_name}"

    items = []

    while True:

        try:

            id = int(input("Enter the ID of the furniture to order: "))

            quantity = int(input("Enter the quantity to order: "))

            items.append({'id': id, 'quantity': quantity})

        except ValueError:

            print("Invalid input. Please enter numeric values for ID and quantity.")

            continue

    more = input("Do you want to order more items? (yes/no): ")

    if more.lower() != 'yes':

        break
```

```
order_invoice = order_from_manufacturer(details, items)

print(f"Order invoice generated: {order_invoice}")

elif choice == '3':

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

    details = f"Customer: {customer_name}"

    items = []

    while True:

        try:

            id = int(input("Enter the ID of the furniture to sell: "))

            quantity = int(input("Enter the quantity to sell: "))

            items.append({'id': id, 'quantity': quantity})

        except ValueError:

            print("Invalid input. Please enter numeric values for ID and quantity.")

            continue

    more = input("Do you want to sell more items? (yes/no): ")

    if more.lower() != 'yes':

        break

    sale_invoice = sell_to_customer(details, items)

    print(f"Sale invoice generated: {sale_invoice}")
```

```
elif choice == '4':

    print("Thank you for using BRJ Furniture Store Management System.
Goodbye!")

    break

else:

    print("Invalid choice. Please try again.")

# Ask if the user wants to choose more options

more_options = input("Do you want to choose from BRJ management store
(yes/no): ")

if more_options.lower() != 'yes':

    print("Thank you for using BRJ Furniture Store Management System.
Goodbye!")

    break

if __name__ == "__main__":

    main()
```

9.2) Operation.py

```
import datetime

from read import read_inventory

from write import write_inventory

def generate_invoice(transaction_type, details, items, inventory, shipping_cost):

    try:
```

```
now = datetime.datetime.now()

timestamp = now.strftime("%Y%m%d_%H%M%S")

filename = f"{transaction_type}_invoice_{timestamp}.txt"

with open(filename, 'w') as file:

    file.write(f"{transaction_type.capitalize()} Invoice\n")

    file.write(f>Date and Time: {now.strftime('%Y-%m-%d %H:%M:%S')}\n")

    file.write(f>Details: {details}\n\n")

total_amount = 0

file.write("Items:\n")

for item in items:

    id = item['id']

    quantity = item['quantity']

    if id not in inventory:

        raise ValueError(f>Error: Furniture ID {id} does not exist in the inventory.")

    inventory_item = inventory[id]

    price = inventory_item['price']

    amount = quantity * price

    total_amount += amount
```

```
file.write(f"ID: {id}, Manufacturer: {inventory_item['manufacturer']], "
          f"Product: {inventory_item['product_name']], Quantity: {quantity}, "
          f"Price: ${price:.2f}, Amount: ${amount:.2f}\n")

vat = total_amount * 0.13

total_amount_with_vat = total_amount + vat

total_amount_with_shipping = total_amount_with_vat + shipping_cost

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

file.write(f"\nTotal Amount: ${total_amount:.2f}\n")

file.write(f"VAT (13%): ${vat:.2f}\n")

file.write(f"Total Amount with VAT: ${total_amount_with_vat:.2f}\n")

file.write(f"Shipping Cost: ${shipping_cost:.2f}\n")

file.write(f"Total Amount to be Paid: ${total_amount_with_shipping:.2f}\n")

return filename

except Exception as e:

    print(f"An error occurred while generating the invoice: {e}")
```

```
    return None
```

```
def get_shipping_cost():
```

```
    while True:
```

```
        try:
```

```
            shipping_cost = float(input("Enter the shipping cost: $"))
```

```
            if shipping_cost <= 0:
```

```
                raise ValueError("Shipping cost must be greater than 0.")
```

```
            return shipping_cost
```

```
        except ValueError as e:
```

```
            print(e)
```

```
def order_from_manufacturer(details, items):
```

```
    try:
```

```
        inventory = read_inventory()
```

```
        shipping_cost = get_shipping_cost() # Get the validated shipping cost from the  
user
```

```
    for item in items:
```

```
        id = item['id']
```

```
        quantity = item['quantity']
```

```
        if id in inventory:
```

```
            inventory[id]['quantity'] += quantity
```

```
        else:
```



```
        print(f"Error: Furniture ID {id} does not exist.")

    write_inventory(inventory)

    return generate_invoice('order', details, items, inventory, shipping_cost)

except Exception as e:

    print(f"An error occurred during the order process: {e}")

    return None


def sell_to_customer(details, items):

    try:

        inventory = read_inventory()

        shipping_cost = get_shipping_cost() # Get the validated shipping cost from the
user

        for item in items:

            id = item['id']

            quantity = item['quantity']

            if id in inventory:

                if inventory[id]['quantity'] >= quantity:

                    inventory[id]['quantity'] -= quantity

                else:

                    print(f"Error: Not enough stock for Furniture ID {id}.")

            else:
```

```
    print(f"Error: Furniture ID {id} does not exist.")

    write_inventory(inventory)

    return generate_invoice('sale', details, items, inventory, shipping_cost)

except Exception as e:

    print(f"An error occurred during the sales process: {e}")

    return None
```

9.3) Read.py

```
def read_inventory(filename='furniture.txt'):
```

```
    inventory = {}
```

```
    with open(filename, 'r') as file:
```

```
        for line in file:
```

```
            parts = line.strip().split(' ')
```

```
            id = int(parts[0])
```

```
            manufacturer = parts[1]
```

```
            product_name = parts[2]
```

```
            quantity = int(parts[3])
```

```
            price = float(parts[4].strip("$"))
```

```
            inventory[id] = {
```

```
                'manufacturer': manufacturer,
```

```
                'product_name': product_name,
```

```
        'quantity': quantity,  
        'price': price  
    }  
  
    return inventory
```

9.4) Write.py

```
def write_inventory(inventory, filename='furniture.txt'):  
  
    with open(filename, 'w') as file:  
  
        for id, details in inventory.items():  
  
            line = f"{id}, {details['manufacturer']}, {details['product_name']},  
{details['quantity']}, ${details['price']:.2f}\n"  
  
            file.write(line)
```

10) Reference

References

altexSoft, 2024. *altexSoft*. [Online]

Available at: <https://www.altexsoft.com/blog/data-structure/>

[Accessed 11 08 2024].

Anon., 2023. *Testbook*. [Online]

Available at: <https://testbook.com/computer-awareness/computer-fundamentals>

[Accessed 27 07 2024].

Anon., 2024. *Nanyang Technological*. [Online]

Available at: <https://www3.ntu.edu.sg/home/assourin/Teaching/foc.htm>

[Accessed 27 07 2024].

Anon., 2024. *Weebly*. [Online]

Available at:

https://ashishmodi.weebly.com/uploads/1/8/9/7/18970467/computer_fundamental.pdf

[Accessed 27 07 2024].

Computer hope, 2024. *Computer hope*. [Online]

Available at: <https://www.computerhope.com/jargon/d/drawio.htm>

[Accessed 27 07 2024].

DataCamp, 2020. *DataCamp*. [Online]

Available at: <https://www.datacamp.com/tutorial/python-IDLE>

[Accessed 25 07 2024].

Economic Times, 2024. *Economic Times*. [Online]

Available at: <https://economictimes.indiatimes.com/definition/pseudocode>

[Accessed 29 07 2024].

Geeksforgeeks, 2024. *Geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/introduction-to-python/>

[Accessed 23 07 2024].

Geeksforgeeks, 2024. *Geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/an-introduction-to-flowcharts/>

[Accessed 10 08 2024].

Geeksforgeeks, 2024. *Geeksforgeeks*. [Online]

Available at: <https://www.geeksforgeeks.org/difference-between-linear-and-non-linear-data-structures/>

[Accessed 11 08 2024].

<https://www.datacamp.com/tutorial/python-IDLE>, 2020. *DataCamp*. [Online]

Available at: <https://www.datacamp.com/tutorial/python-IDLE>

[Accessed 25 07 2024].

javatPoint, 2024. *javatPoint*. [Online]

Available at: <https://www.javatpoint.com/data-structure-introduction>

[Accessed 11 08 2024].

MyEducator, 2024. *MyEducator*. [Online]

Available at: <https://app.myeducator.com/reader/web/1204b/lesson1/mx1ix/>

[Accessed 27 07 2024].

techTarget, 1999. *techTarget*. [Online]

Available at:

<https://www.techtarget.com/whatis/definition/algorithm#:~:text=An%20algorithm%20is%20a%20procedure,%2D%20or%20software%2Dbased%20routines.>

[Accessed 29 07 2024].

techTarget, 2005. *techTarget*. [Online]

Available at: <https://www.techtarget.com/searchdatamanagement/definition/data-structure#:~:text=Data%20structures%20define%20the%20organization,character%20strings%20used%20as%20tags.>

[Accessed 11 08 2024].

Testbook, 2023. *Testbook*. [Online]

Available at: <https://testbook.com/computer-awareness/computer-fundamentals>

[Accessed 27 07 2024].

Tutorials Point, 2024. *Tutorials Point*. [Online]

Available at:

[https://www.tutorialspoint.com/data_structures_algorithms/data_structures_and_types.h
tm](https://www.tutorialspoint.com/data_structures_algorithms/data_structures_and_types.htm)

[Accessed 11 08 2024].

UnStop, 2024. *UnStop*. [Online]

Available at: [https://unstop.com/blog/difference-between-linear-and-non-linear-data-
structure](https://unstop.com/blog/difference-between-linear-and-non-linear-data-structure)

[Accessed 11 08 2024].

UnStop, 2024. *UnStop*. [Online]

Available at: [https://unstop.com/blog/difference-between-linear-and-non-linear-data-
structure](https://unstop.com/blog/difference-between-linear-and-non-linear-data-structure)

[Accessed 11 08 2024].

11 Report

Originality report

COURSE NAME CS4051NT_FOC

STUDENT NAME

Purnima Basnet

FILE NAME

Coursework Fundamental(ajayraj)

(1)

REPORT CREATED Aug

16, 2024

Summary

Flagged passages	9	2%
Cited/quoted passages	5	0.8%

Web matches

geeksforgeeks.org	5	0.8%
cliffsnotes.com	1	0.6%
myeducator.com	1	0.2%
optimizely.com	2	0.2%
iarj.in	1	0.2%
datacamp.com	1	0.2%
techtarget.com	1	0.2%
studyx.ai	1	0.1%

chegg.com

1

0.1%

1 of 14 passages**Student passage** FLAGGED

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...

Top web match

Advanced Database CC6001NT Advanced Database System Development 40% Individual Coursework Student

Name: Susan Shrestha London Met ID: 20048536 College ID: np05cp4s210004@iic.edu.np Assignment Due...

20048536 Susan Shrestha (docx) - CliffsNotes <https://www.cliffsnotes.com/study-notes/14093778>

2 of 14 passages

Student passage CITED

...indenting text. Syntax mistakes can result from improper indentation. **Python is a programming language that facilitates faster work and more effective system integration.** (Geeksforgeeks, 2024).

Top web match

Python is a programming language that facilitates faster work and more effective system integration.

Python comes in two main versions: Python 2 and Python 3.

An Overview on Python Programming

<https://www.iarj.in/index.php/ijrase/article/download/276/311/620>

3 of 14 passages**Student passage** FLAGGED

While the name IDLE was chosen as a tribute to Eric Idle, one of the founding members of the British comic group Monty Python, Guido Van...

Top web match

The story behind the name IDLE is similar to Python. Guido Van Rossum named Python after the British comedy group Monty Python **while the name IDLE was chosen to pay tribute to Eric Idle**, who was **one...**

Introduction to Python IDLE Tutorial - DataCamp <https://www.datacamp.com/tutorial/python-IDLE>

4 of 14 passages

Student passage **FLAGGED**

you will also pick up **more** sophisticated capabilities **that will** improve **your** ability to write **and** create **documents, reports, articles, newsletters, flyers, and letters**

Top web match

In these lessons, **you will** learn not only the basic features of Word but **also more** advanced features **that will** enhance **your** skills in creating **and** writing **documents, reports, articles, newsletters,...**

1.1 Introduction to Microsoft Word - MyEducator
<https://app.myeducator.com/reader/web/1204b/lesson1/mx1ix/>

5 of 14 passages

Student passage **FLAGGED**

...Prompt **the user to choose one of the following options**

Top web match

Write a program to prompt a user to enter a number. Then display a menu of choices for **the user to** select as follows: Please **choose one of the following options**: Print the number in binary separating...

Solved Write a program to prompt a user to enter a number. - Chegg
<https://www.chegg.com/homeworkhelp/questions-and-answers/write-program-prompt-user-enter-number-display-menu-choices-user-selectfollows-please-cho-q12268777>

6 of 14 passages

Student passage QUOTED

...PRINT "Shipping cost must be greater than 0.

Top web match

Amount <= 0) { throw new ValidationException("Shipping cost must be greater than 0"); } }
protected override void ValidateShippingItemTotal(Money money) { if (money.Amount <= 0) { throw
new...

Shipping calculator - Optimizely <https://docs.developers.optimizely.com/customized-commerce/v13.0.0commerce-cloud/docs/calculating-orders-shipping-calculator>

7 of 14 passages

Student passage FLAGGED

It makes use of symbols that are linked together to represent the flow of information and processing. The process of creating a flowchart for an algorithm is known as "flowcharting"

Top web match

It makes use of symbols which are connected among them to indicate the flow of information and processing. The process of drawing a flowchart for an algorithm is known as "flowcharting".

An introduction to Flowcharts - GeeksforGeeks <https://www.geeksforgeeks.org/an-introduction-to-flowcharts/>

8 of 14 passages

Student passage FLAGGED

...is known as "flowcharting". Basic Symbols for Flowchart Designs :**Terminal: The oval symbol represents**

Start, Stop, and Halt in a program's logic flow

Top web match

Terminal: The oval symbol indicates Start, Stop and Halt in a program's logic flow. ... Arrows represent the direction of flow of control ...

An introduction to Flowcharts - GeeksforGeeks <https://www.geeksforgeeks.org/an-introduction-to-flowcharts/>

9 of 14 passages

Student passage FLAGGED

...commonly employed in program logic under certain error situations. **Terminal is the first and last symbols in the flowchart.**

Top web match

Terminal: The oval symbol indicates Start, Stop and Halt in a program's logic flow. A pause/halt is generally used in a program logic under some error conditions. **Terminal is the first and last...**

An introduction to Flowcharts - GeeksforGeeks <https://www.geeksforgeeks.org/an-introduction-to-flowcharts/>

10 of 14 passages

Student passage CITED

Processing: Each block represents an arithmetic operation. **All arithmetic operations, such as adding, subtracting, multiplying, and dividing, are represented by an action or process symbol**

Top web match

Processing: A box represents arithmetic instructions. **All arithmetic processes such as adding, subtracting, multiplication and division are indicated by action or process symbol.**

An introduction to Flowcharts - GeeksforGeeks <https://www.geeksforgeeks.org/an-introduction-to-flowcharts/>

11 of 14 passages

Student passage CITED

in computer science and computer programming with the intention of storing **data** for use with different algorithms; these combinations of **data** structures and **algorithms** are known **as data structures and...**

Top web match

In **computer science and computer programming**, a **data** structure might be selected or designed to store **data** for the purpose of using it with various **algorithms** -- commonly referred to as **data**...

What is a data structure? | Definition

TechTarget <https://www.techtarget.com/searchdatamanagement/definition/data-structure>

12 of 14 passages

Student passage **FLAGGED**

A **data structure** that stores information either **sequentially or linearly** is called a **linear data structure**. Data is structured in a linear data structure so...

Top web match

Linear Data Structure: **Data structure** where data elements are arranged **sequentially or linearly** where each and every element is attached to its previous and next adjacent is called a **linear data**...

Introduction to Linear Data Structures - GeeksforGeeks <https://www.geeksforgeeks.org/introduction-to-lineardata-structures/>

13 of 14 passages

Student passage **FLAGGED**

Elements in an array are kept in **memory** at **contiguous locations**, meaning they are always **next to** one another. It is homogenous, which means that only...

Top web match

Array **elements** are stored in **memory** in **contiguous locations**, meaning they are placed **next to** each other in memory. Final Answer. The correct ...

Choose the correct statement Choose the - StudyX <https://studyx.ai/homework/100896787-choose-the-correctstatement-choose-the-correct-option-an-array-is-a-group-of-elements-of>

14 of 14 passages

Student passage CITED

...raise ValueError(**"Shipping cost must be greater than 0."**)

Top web match

```
Amount <= 0) { throw new ValidationException("Shipping cost must be greater than 0"); } }  
protected override void ValidateShippingItemTotal(Money money) { if (money.Amount <= 0) { throw  
new...
```

Shipping calculator - Optimizely <https://docs.developers.optimizely.com/customized-commerce/v13.0.0commerce-cloud/docs/calculating-orders-shipping-calculator>
