Name: Abhishek Sharma

Section: L1C9

Table of Contents

[1 Introduction 5](#_Toc172115238)

[1.1 Introduction to Python 5](#_Toc172115239)

[1.2 Introduction to IDLE 6](#_Toc172115240)

[1.3 Introduction to coursework 6](#_Toc172115241)

[1.4 Goals and Objectives 7](#_Toc172115242)

[2 Algorithm 8](#_Toc172115243)

[3 Pseudocode 8](#_Toc172115244)

[4 Flowchart 8](#_Toc172115245)

[5 Data Structures 9](#_Toc172115246)

[6 Program 10](#_Toc172115247)

[7 Testing 10](#_Toc172115248)

[8 Conclusion 11](#_Toc172115249)

[9 References 12](#_Toc172115250)

[10 Appendix 13](#_Toc172115251)

x**Table of Figures:**

[Figure 1: Python Logo 5](#_Toc172115079)

[Figure 2: IDLE Logo 6](#_Toc172115080)

**Table of Tables:**

# Introduction

The "Fundamentals of Computing" module (CS4051NT), led by Mr. Ajayraj Bhattrai, focuses on Python programming, and covers essential computing concepts including, algorithms, flowcharts, and software development practices. This semester-long course assesses our performance through coursework, which accounts for 60% of our grade, and an in-course test, making up the remaining 40%. Designed to build our foundational knowledge and enhance critical thinking skills, the module combines lectures, tutorials, workshops, practical assignments, and coursework to provide a comprehensive learning experience in computing.

## Introduction to Python

Python is an admired general purpose; high level language with multifunctional uses and is acknowledged for its ease of use. Standardized by Guido van Rossum in 1991 and consistently developed by the Python Software Foundation, Python was designed and fine-tuned to be as readable and understandable, with simple and elegant syntactic structure, which would let the programmers express any idea in a clean, concise manner. Another significant aspect of Python is that it enables the quick development cycle where the members deliver fast, and it is amenable to integration. Over time, Python has evolved into two major versions: Two main groups of the programming language: Python 2 and Python 3 which differ in the features they provide, the capabilities they offer and the paradigms they support. However, both versions are still in use in multiple spheres ranging from web development, scientific computing, AI, and automation. (Geeks For Geeks, 2024)

A blue and yellow snake logo

Description automatically generated

Figure 1: Python Logo

## Features of Python

**Free and Open Source**

Python is an open-source language that can be downloaded from the official Python website and can be downloaded by just clicking on the Download Python link. Since it is an open source, the code is easily accessible to everyone for making modifications. It is free from legal protection and can be downloaded, used, and shared with anyone and everyone.

**Easy to Code**

Python is a great programming language that is relatively easier to learn than languages such as C, C#, JavaScript or Java. It will take you a few hours or days to learn how to program in Python if you are a beginner. It is intended for new users as well as advanced developers in the field and network engineering.

**Easy to Read**

Python is easy to learn since its structures are quite basic and uncluttered. Thus, the code structure in Python is very clear, and instead of semicolons or brackets, the code blocks are identified by indentations.

**Object-Oriented Language**

Python also supports Object-Oriented Programming which comprises of the notions of classes and object encapsulation. Due to this, it is an ideal option for creating intricate applications and programs.

**GUI Programming Support**

There are GUI modules available for Python like PyQt5 and PyQt4, wxPython, Tkinter and so on. In the present scenario, GUI frameworks are widely used, and out of all, PyQt5 is in great demand for developing graphical applications. And many more. (Geeks for geeks, 2024)

## Introduction to IDLE

IDLE, Interactive Development and Learning Environment for Python runs on Windows Unix as well as MacOS. It has a feature of a Python shell for entering code and the prompt for input, output, and exceptional conditions are in the colour. There are nice additional options that you can see in the text editor such as undo, highlighter for Python code and tips when writing functions. You can find specific text to replace it with or in files with a particular name. Like many programming languages, IDLE also provides the debugger to track down the errors in your code and allows you to step through your code to see how the program runs. that is convenient to work with adding that it has settings and tools included within. (Python Software Foundation, 2024)

A close-up of a paper

Description automatically generated

Figure 2: IDLE Logo

## Introduction to coursework

This coursework for the "Fundamentals of Computing" module (CS4051NT) focuses on developing a Python program to manage inventory and transactions for a furniture store. The program will read, and update information stored in a text file, displaying available furniture, and generating transaction notes for both purchases from manufacturers and sales to customers. It will incorporate features like error handling, input validation, and the calculation of VAT for sales transactions. The coursework emphasizes modular programming, using appropriate data structures for efficient data management, and includes documentation, algorithm development, and testing to ensure functionality and reliability.

## Goals and Objectives

The objectives of our coursework for the "Fundamentals of Computing" module are designed to achieve comprehensive understanding and practical skills in Python programming and software development. Our goals include:

* **Develop a Functional Inventory Management System:** Create a Python program to manage furniture inventory for a store, allowing updates for sales and orders directly impacting stock levels.
* **Implement Modular Programming:** Structure the program into separate functions for managing inventory, processing transactions, and generating invoices to ensure code reusability and clarity.
* **Utilize Effective Data Structures:** Employ appropriate data structures like dictionaries and lists in Python to efficiently store and manipulate furniture data, ensuring optimal performance for inventory operations.
* **Document and Present:** Provide clear documentation including algorithm descriptions, pseudocode, and flowcharts to illustrate program logic and functionality. Include a testing section to validate program behavior under different scenarios.

These objectives focus on developing practical Python programming skills while addressing the specific requirements of managing furniture inventory and transaction processing as outlined in our coursework.

# Algorithm

# Pseudocode

# Flowchart

# Data Structures

In developing the furniture inventory management system, several data structures in Python are utilized to efficiently manage data storage, input/output operations, and transaction processing:

**Lists**: Used to store the list of furniture items read from and written to the text file. Lists allow for sequential access and modification of data entries, making them suitable for managing dynamic inventory updates and transaction histories.

**Dictionaries**: Employed to map unique identifiers (IDs) of furniture items to detailed information such as manufacturer, product name, available quantity, and price per unit. This facilitates quick look up and retrieval of specific item details during transaction processing and inventory updates.

**Strings**: Utilized for processing and formatting textual data, such as parsing input from administrators or formatting output for transaction receipts. String manipulation functions are employed to ensure data integrity and clarity in generated invoices and transaction notes.

**File Handling**: Python's built-in file handling capabilities are employed for input/output operations, allowing the program to read furniture data from and write updated data back to the text file. This ensures persistent storage and retrieval of inventory information between program sessions.

**Custom Class**: For more complex applications, custom-defined classes can encapsulate furniture item attributes and methods, enhancing code modularity and scalability. Classes can represent furniture items as objects with properties and behaviors tailored to specific application requirements.

Each data structure is chosen based on its suitability for handling various aspects of the inventory management system, from efficient data retrieval and manipulation to ensuring robust file handling and user interaction. (Akash, 2024)

# Program

# Testing

# Conclusion

In conclusion, the construction of the BRJ Furniture Store Inventory Management System has therefore been a worthy exercise in learning Python programming and data processing. It was somewhat difficult but also very rewarding to produce a functional approach to organizing stock, take payments, and issue receipts.  
  
I maintain clean and organized code that is written in the standard manner of software development. I was able to solve problems related to organizing and processing the data of inventory with the assistance of lists and dictionaries. Since the program is capable of reading from and writing text files, data is secured and retrievable for regular use.

I also learned skills in the design of algorithms, error management and file management from this project. I also got to know about the documentation of my work and it’s testing as well for it to be confirmed that it works effectively.  
  
During planning, coding, and debugging stages I obtained the experience in such aspects as team cooperation, critical thinking, and practical use of the learned material.  
  
Therefore, the BRJ Furniture Store Inventory Management System proves my commitment and IT competency. I am glad that I accomplished such things, and I think it will be useful in the store. General, this project has prepared me for future software development related tasks, and I will use the skills acquired in my future learning and working endeavors.

# References

Akash. (2024, July 17). *data-structures-in-python*. Retrieved from edureka: https://www.edureka.co/blog/data-structures-in-python/

Geeks For Geeks. (2024, July 16). *introduction-to-python*. Retrieved from Geeks For Geeks: https://www.geeksforgeeks.org/introduction-to-python/

Geeks for geeks. (2024, July 17). *python-features*. Retrieved from geeksforgeeks: https://www.geeksforgeeks.org/python-features/

Python Software Foundation. (2024, July 16). *idle.html*. Retrieved from docs.Python: https://docs.python.org/3/library/idle.html

# Appendix