



**UTM**  
UNIVERSITI TEKNOLOGI MALAYSIA

Malaysia-Japan  
International  
Institute of Technology  
(MJIT)

UNIVERSITI TEKNOLOGI MALAYSIA




MALAYSIAN JAPAN INTERNATIONAL INSTITUTE OF TECHNOLOGY

**SMJE 4383-01 Advanced Programming**

**2023/2024**

# Assignment 1

---

<b>Group: 3</b> <b>Section: 1</b>			
<b>NAME</b>	OMER WAIL YASSIN SAADELDIEN	MUHAMMAD AFIQ HAKIMI BIN MOHD RAHIME	MD NASIF RAHMAN
<b>MATRIC NO</b>	A20MJ5004	A20MJ0182	A20MJ9050
<b>LECTURER'S NAME:</b> DR. Shahrum Shah Abdulla			

# Table of contents

---

Title Page .....	1
Abstract .....	3
Introduction .....	3
Progress .....	4
Conclusion .....	10

---

## Abstract

---

The main goal of this project is to use Python to collaboratively construct an online bookstore with an emphasis on teamwork, improving programming skills, and efficiently utilising GitHub and Git for version control. The goal of "Booknest," an online bookstore, is to enable visitors to quickly peruse, look up, and buy a wide variety of books. This report gives a summary of the steps taken to put the BookStore class's basic structure into place, load data from external files, and enable important features like book browsing, author or title searches, and book purchases. The codebase shows a basic structure for additional improvement and development.

## Introduction

---

The importance of online platforms for many businesses, including bookstores, in the quickly changing digital landscape cannot be emphasised. Creating an online bookshop in Python is a fun and useful project that may help you learn version control with Git and GitHub, improve your programming abilities, and build collaboration. The goal of the Booknest project is to provide an intuitive and effective platform that will enable users to easily browse a vast selection of books, perform searches using particular parameters, and make purchases.

The project is now in the process of establishing a fundamental class called BookStore, which contains all of the necessary functionality to import names of authors and books from external files, display book information, search for books using user input, and enable book purchases. Because the project places a strong emphasis on using Git and GitHub effectively, collaborative development is well-managed and team member contributions may be integrated with ease.

The implemented functions, the BookStore class's structure, and the use of external data files to store book and author information are all covered in detail in this paper. It will also describe how

to browse, search, and buy books using the user interface. An outline of upcoming development plans, including possible additions, features, and upgrades for the Booknest online bookstore, is provided at the end of the report.

## Progress

---

So far we did around 30% of the total code and the coding is still in progress. And i will show you now what our group did so far.

BookStore Class:

### 1. Initialization Method (`__init__`):

The constructor for the BookStore class is the `__init__` method. Two qualities are initialised: books and authors.

authors: Author names are read from the "author\_names.txt" file by using the `load_authors` method.

books: The "book\_names.txt" file contains book titles and quantities, which are read by the `load_books` method.

### 2. Loading Authors (`load_authors`): This function retrieves the list of authors by reading their names from the "author\_names.txt" file.

### 3. The "`load_books`" function retrieves book titles and quantities from the "book\_names.txt" file while loading books.

Creates a dictionary (books) with the book IDs serving as the keys and the title, author, price, and quantity of each book serving as the values.

uses the matching author for every book; in the event that no author is found, the default is 'Unknown Author'.

### 4. Browse Books (`browse_books`): This function iterates over the book lexicon and outputs details about every book, such as ID, title, author, cost, and quantity.

5. Searching Books (search\_books): This function loops over the books dictionary using a search word as input. It outputs information about the matching books if the search word appears in the author or title (case insensitive).

6. Buying Book (purchase\_book): This function accepts a book ID as an input. Verifies whether the chosen book is available (quantity > 0). In that case, a purchase confirmation is printed and the quantity is lowered by 1. If not, it says that the book is not available.

Moving on to the **Main programme**:

Prints a welcome message for the Booknest online retailer and initialises it.  
initialises the store variable and creates an instance of the BookStore class.

**User Interaction Loop** (while True): Asks the user to select between making a purchase and leaving.

**Buy Action** ({if action.lower() == 'buy':): Calls the browse\_books method to print the list of books that are available.

requests that the user input the book's ID in order to complete the purchase.

**Search and Purchase**: Verifies the validity of the ID entered.

If it is valid, it calls the search\_books method to look for the selected book and then the purchase\_book method to try to buy it.

**Leave Action** ({elif action.lower() == 'leave':): This programme ends by printing a farewell message and exiting the loop.

With functionality for data loading, book browsing, searching, and purchasing, the code lays the groundwork for an online bookstore. It offers an intuitive user interface and lays the groundwork for future advancements. Collaborative development and version control are supported by the use of external files for data storage and the focus on Git and GitHub integration.

The code is available at our repository in github:

[https://github.com/YoungMars3/Python\\_Group3](https://github.com/YoungMars3/Python_Group3)

### The code for our Project:

# Defining a class named BookStore

class BookStore:

def \_\_init\_\_(self):

self.authors = self.load\_authors()

self.books = self.load\_books()

# Reading author names from "author\_names.txt"

def load\_authors(self):

with open('author\_names.txt', 'r') as file:

return [line.strip() for line in file]

# Reading book names from "book\_names.txt"

def load\_books(self):

# Load book names and quantities from the file

with open('book\_names.txt', 'r') as file:

book\_data = [line.strip().split(',') for line in file]

# Create a dictionary with book IDs, names, and quantities

books = {}

for i, (name, quantity\_str) in enumerate(book\_data, start=1):

# Use the corresponding author for each book

author = self.authors[i - 1] if i <= len(self.authors) else 'Unknown Author'

books[str(i)] = {'title': name, 'author': author, 'price': 10 + i, 'quantity': int(quantity\_str)}

return books

# Displaying info bout all the books in directory

def browse\_books(self):

for id, book in self.books.items():

print(f'ID: {id}, Title: {book['title']}, Author: {book['author']}, Price: {book['price']},

Quantity: {book['quantity']}")

```

# Looking for search terms
def search_books(self, search_term):
    for id, book in self.books.items():
        if search_term.lower() in book['title'].lower() or search_term.lower() in
book['author'].lower():
            print(f'ID: {id}, Title: {book['title']}, Author: {book['author']}, Price: {book['price']},
Quantity: {book['quantity']}")

# Checking for availability and purchasing book
def purchase_book(self, id):
    if self.books[id]['quantity'] > 0:
        self.books[id]['quantity'] -= 1
        print(f'\nYou have purchased {self.books[id]['title']} by {self.books[id]['author']}'.)
    else:
        print("\nSorry, this book is out of stock.")

# Output starts from here:
print("Welcome to Booknest.")
store = BookStore()

# Choosing whether to buy or leave
while True:
    action = input("\nWhat would you like to do? (Buy/Leave): ")

#Printing names of all the books with details
if action.lower() == 'buy':
    print("\nFollowing is the list of books available:")
    store.browse_books()

```

```
#Prompt to enter the ID number of a book
idbuy = input("\nEnter ID number of book you want to buy: ")

if idbuy in store.books:
    store.search_books(store.books[idbuy]['title'])
    store.purchase_book(idbuy)
else:
    print("\nInvalid ID. Please choose a valid book ID.")

elif action.lower() == 'leave':
    print("\nThank you. Please come again")
    break
```

The code also has some text files that need to be downloaded along with the code to work. All details available at the read me file in our repository.



# BOOKSTORE

Including the report, the Team members were each handed a phase and checked in for each other

## CODING PHASE 1

Making the searching feature. Search: Give consumers the option to look up books by author or title.



## CODING PHASE 2

Shopping Cart: Customers can put books in their carts and check out.



## CODING PHASE 3

User Registration: Put in place login and user registration functionality.



## CODING PHASE 4

Purchasing: The application should handle processing payments so that users can

# BOOKSTORE TO BE DONE

## STILL IN PROGRESS

We are in the making of the rest of the phases

## CODING PHASE 5

Choosing a hosting service and setting up an account and preparing the application



## CODING PHASE 6

Ready to deploy and configure the database and storage and testing the deployment



## CODING PHASE 7

We custom the domain and make security and monitoring



## CODING PHASE 8

Lastly implement the back up and recovery and document the whole process and support the website created

## Conclusion

---

In a nutshell, our team has learned a lot and been productive during the joint development of the Booknest online bookshop project. Each of Omer, Nasif, and Afîq was essential to the project's success. Afîq concentrated on Git and GitHub integration, Nasif supplied thorough code documentation, and Omer added further capabilities. The accompanying flowcharts illustrate how organised and well-coordinated the team's collaboration was.

The project not only accomplished its original objectives of improving programming abilities and grasping version control, but it also laid a strong basis for further advancement. Nasif has provided clear and understandable documentation for the codebase, while Afîq has set up Git and GitHub to facilitate organised and cooperative development.

In the long run, the Booknest project marks a turning point in our group's education and development. It is evidence of our ability to work together as a team and of our individual skill. We are sure that the knowledge and abilities we have gained will help make future endeavours successful as we work to improve the online bookshop. The Booknest project is a prime example of how collaboration, skill building, and efficient project management work together.