UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+
ACCREDITED UNIVERSITY

Experiment 1.4

Student Name: Gaurav Kumar **UID: 22MCC20177**

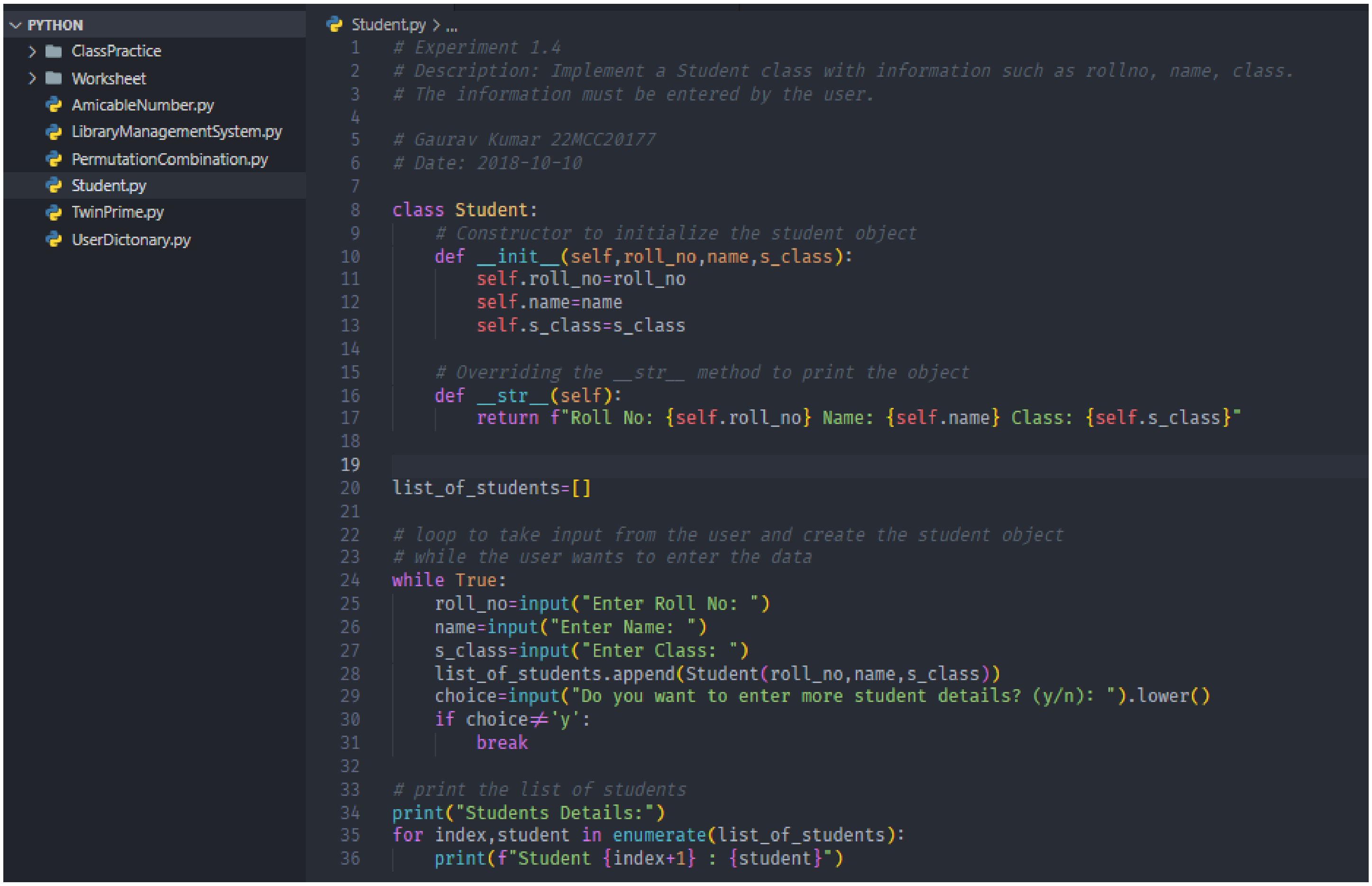
Branch: CC-DEVOPS **Section/Group : 1/B**

Semester: 1 **Date of Performance: 28/Oct/2022**

Subject Name : Python Programming **Subject Code: 22CAH645**

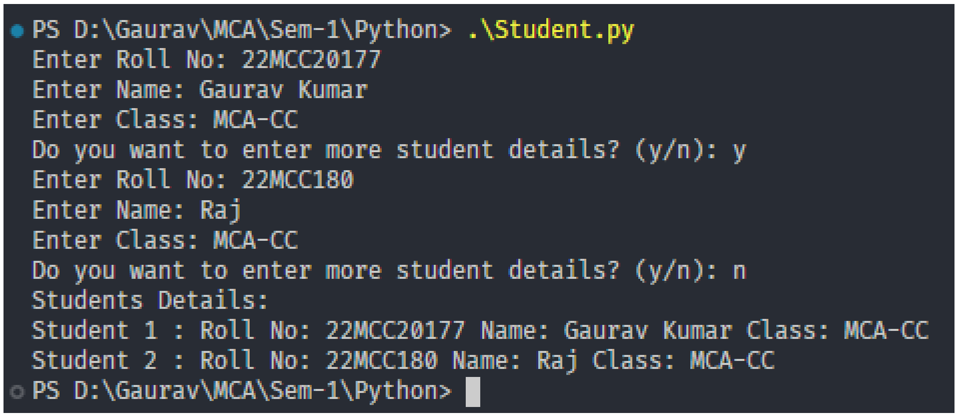
1) **Task to be done: Implement a Student class with information such as roll no, name, and class. The information must be entered by the user.**

Steps for experiment/practical:



```python
# Experiment 1.4
# Description: Implement a Student class with information such as rollno, name, class.
# The information must be entered by the user.

# Gaurav Kumar 22MCC20177
# Date: 2018-10-10

class Student:
    # Constructor to initialize the student object
    def __init__(self,roll_no,name,s_class):
        self.roll_no=roll_no
        self.name=name
        self.s_class=s_class

    # Overriding the __str__ method to print the object
    def __str__(self):
        return f"Roll No: {self.roll_no} Name: {self.name} Class: {self.s_class}"


list_of_students=[]

# loop to take input from the user and create the student object
# while the user wants to enter the data
while True:
    roll_no=input("Enter Roll No: ")
    name=input("Enter Name: ")
    s_class=input("Enter Class: ")
    list_of_students.append(Student(roll_no,name,s_class))
    choice=input("Do you want to enter more student details? (y/n): ").lower()
    if choice!='y':
        break

# print the list of students
print("Students Details:")
for index,student in enumerate(list_of_students):
    print(f"Student {index+1} : {student}")
```

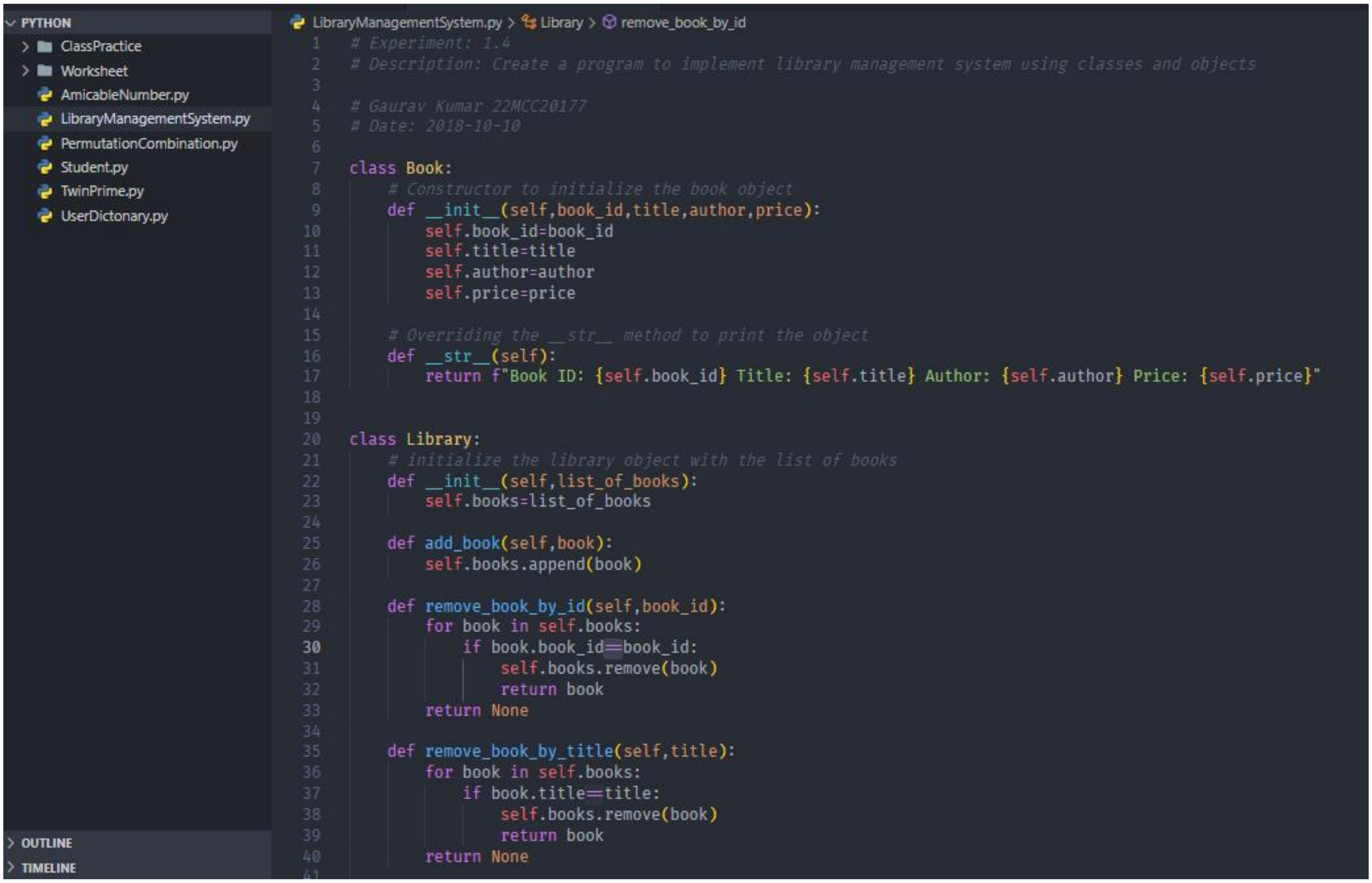
Output (screenshots):



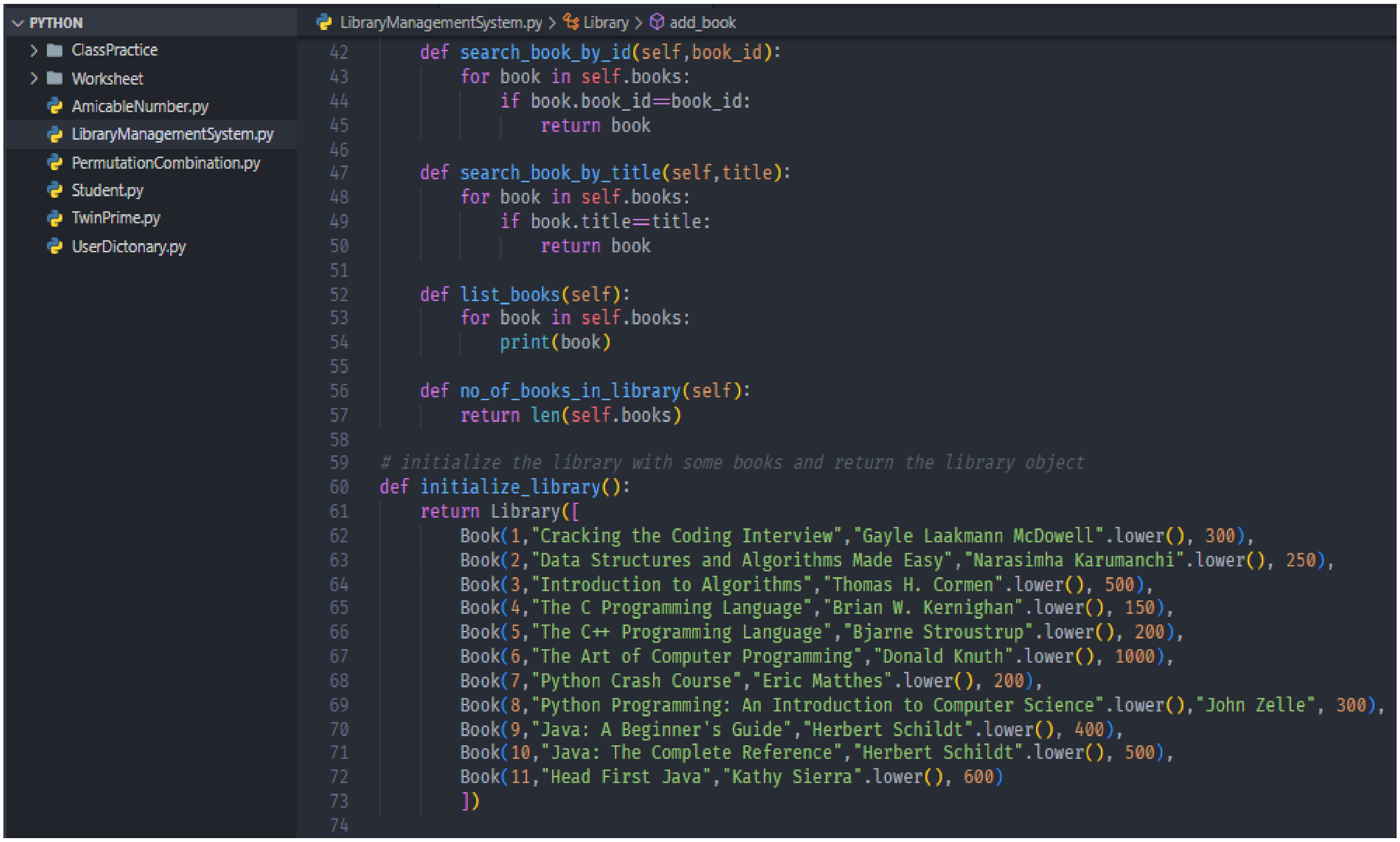
```
PS D:\Gaurav\MCA\Sem-1\Python> .\Student.py
Enter Roll No: 22MCC20177
Enter Name: Gaurav Kumar
Enter Class: MCA-CC
Do you want to enter more student details? (y/n): y
Enter Roll No: 22MCC180
Enter Name: Raj
Enter Class: MCA-CC
Do you want to enter more student details? (y/n): n
Students Details:
Student 1 : Roll No: 22MCC20177 Name: Gaurav Kumar Class: MCA-CC
Student 2 : Roll No: 22MCC180 Name: Raj Class: MCA-CC
PS D:\Gaurav\MCA\Sem-1\Python>
```

2) **Task to be done: Create a program to implement library management system using classes and objects.**

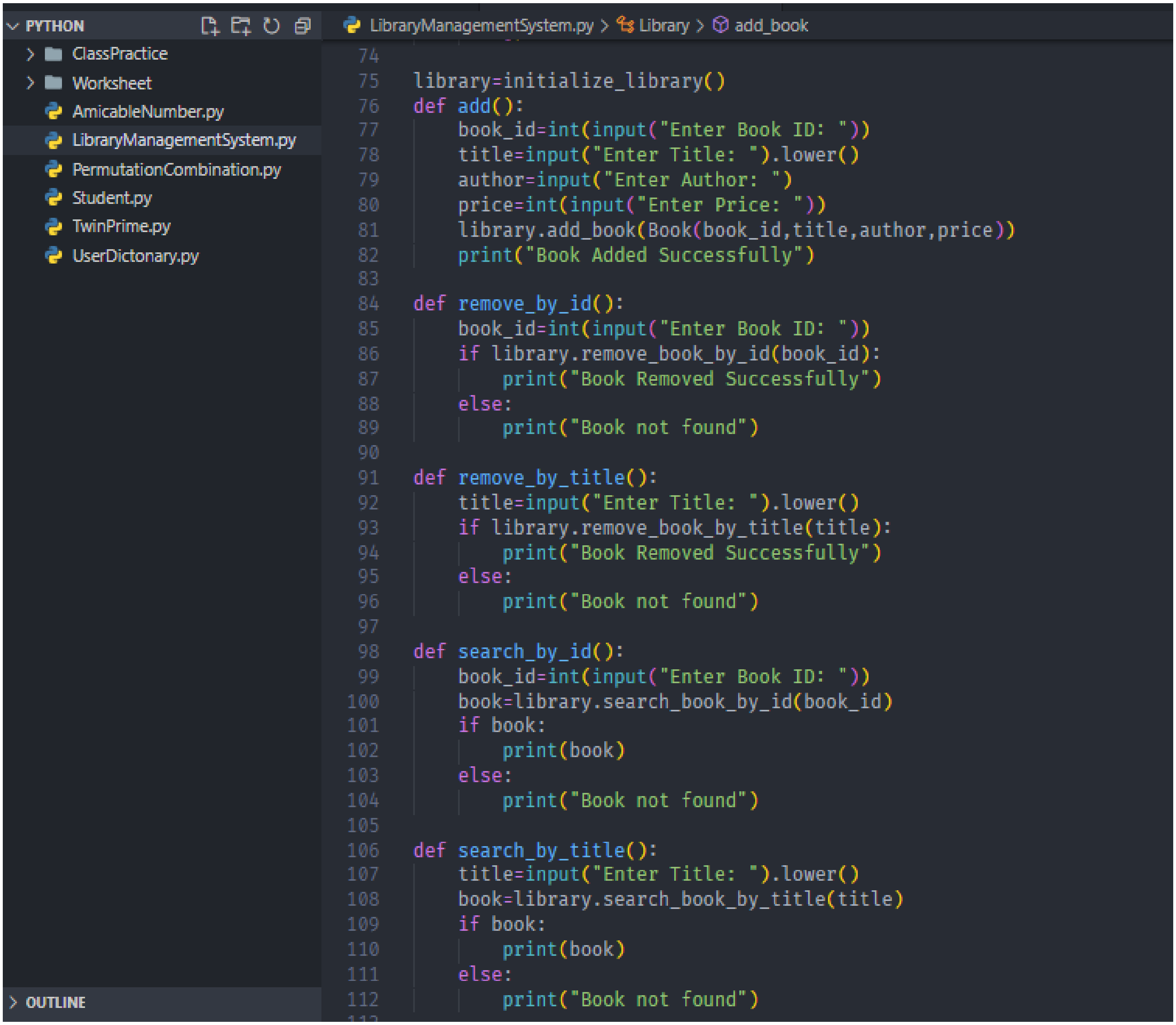
Steps for experiment/practical:


PYTHON
ClassPractice
Worksheet
AmicableNumber.py
LibraryManagementSystem.py
PermutationCombination.py
Student.py
TwinPrime.py
UserDictonary.py
OUTLINE
TIMELINE
LibraryManagementSystem.py > Library > remove_book_by_id

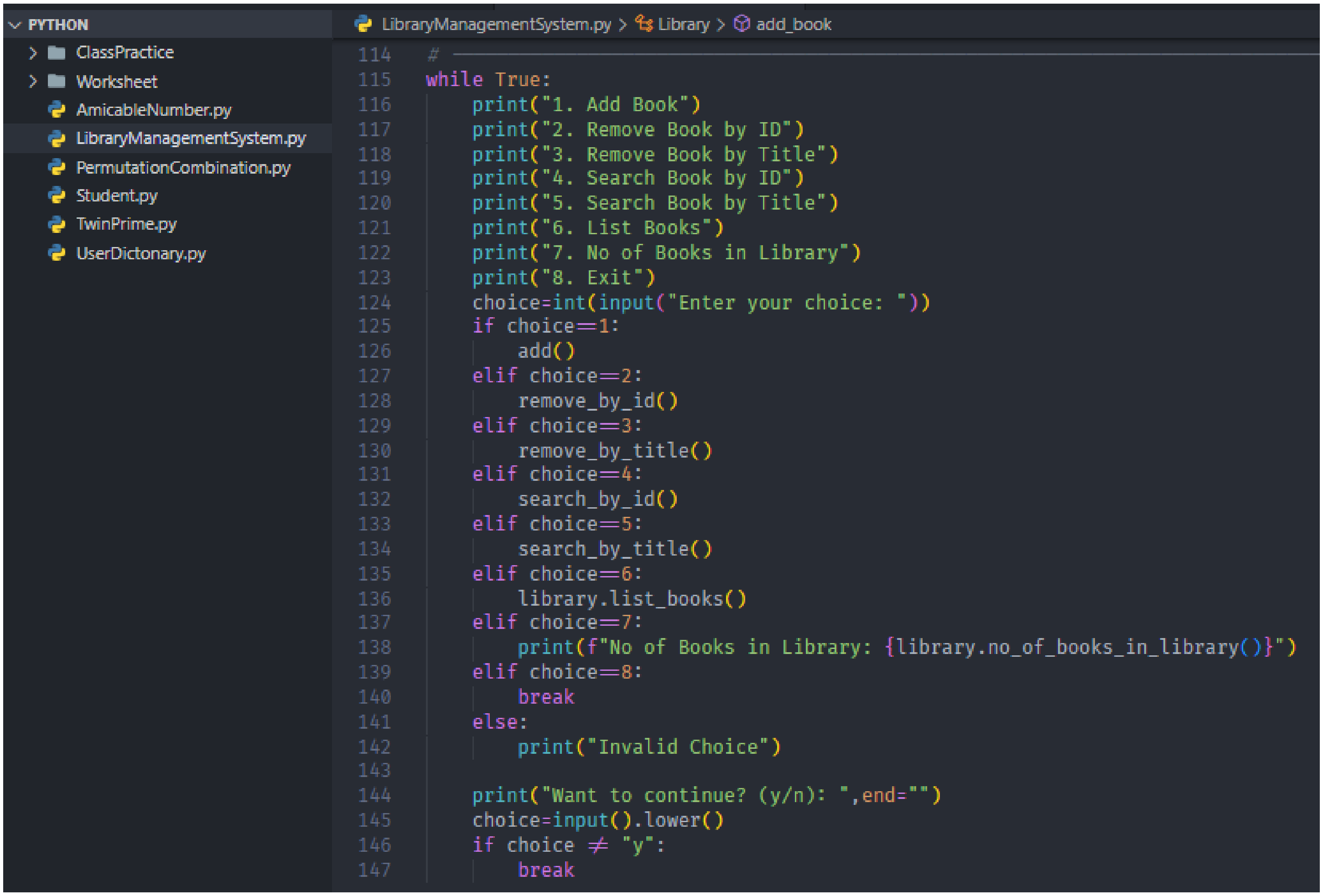
```python
# Experiment: 1.4
# Description: Create a program to implement library management system using classes and objects

# Gaurav Kumar 22MCC20177
# Date: 2018-10-10

class Book:
    # Constructor to initialize the book object
    def __init__(self,book_id,title,author,price):
        self.book_id=book_id
        self.title=title
        self.author=author
        self.price=price

    # Overriding the __str__ method to print the object
    def __str__(self):
        return f"Book ID: {self.book_id} Title: {self.title} Author: {self.author} Price: {self.price}"


class Library:
    # initialize the library object with the list of books
    def __init__(self,list_of_books):
        self.books=list_of_books

    def add_book(self,book):
        self.books.append(book)

    def remove_book_by_id(self,book_id):
        for book in self.books:
            if book.book_id==book_id:
                self.books.remove(book)
                return book
        return None

    def remove_book_by_title(self,title):
        for book in self.books:
            if book.title==title:
                self.books.remove(book)
                return book
        return None
```


PYTHON
ClassPractice
Worksheet
AmicableNumber.py
LibraryManagementSystem.py
PermutationCombination.py
Student.py
TwinPrime.py
UserDictonary.py
LibraryManagementSystem.py > Library > add_book

```python
    def search_book_by_id(self,book_id):
        for book in self.books:
            if book.book_id==book_id:
                return book

    def search_book_by_title(self,title):
        for book in self.books:
            if book.title==title:
                return book

    def list_books(self):
        for book in self.books:
            print(book)

    def no_of_books_in_library(self):
        return len(self.books)

# initialize the library with some books and return the library object
def initialize_library():
    return Library([
        Book(1,"Cracking the Coding Interview","Gayle Laakmann McDowell".lower(), 300),
        Book(2,"Data Structures and Algorithms Made Easy","Narasimha Karumanchi".lower(), 250),
        Book(3,"Introduction to Algorithms","Thomas H. Cormen".lower(), 500),
        Book(4,"The C Programming Language","Brian W. Kernighan".lower(), 150),
        Book(5,"The C++ Programming Language","Bjarne Stroustrup".lower(), 200),
        Book(6,"The Art of Computer Programming","Donald Knuth".lower(), 1000),
        Book(7,"Python Crash Course","Eric Matthes".lower(), 200),
        Book(8,"Python Programming: An Introduction to Computer Science".lower(),"John Zelle", 300),
        Book(9,"Java: A Beginner's Guide","Herbert Schildt".lower(), 400),
        Book(10,"Java: The Complete Reference","Herbert Schildt".lower(), 500),
        Book(11,"Head First Java","Kathy Sierra".lower(), 600)
        ])

```

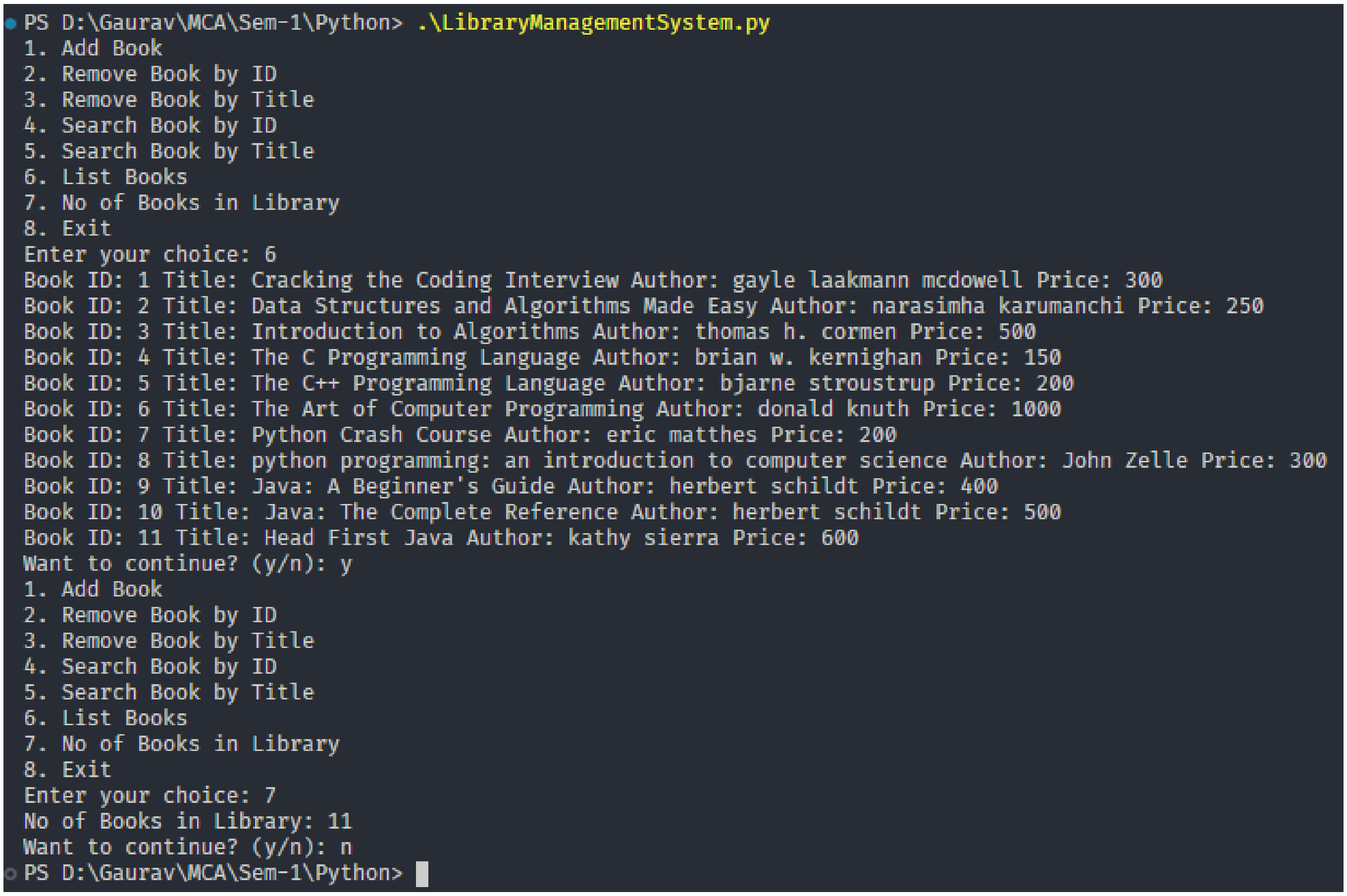

PYTHON
ClassPractice
Worksheet
AmicableNumber.py
LibraryManagementSystem.py
PermutationCombination.py
Student.py
TwinPrime.py
UserDictonary.py
OUTLINE
LibraryManagementSystem.py > Library > add_book

```python

library=initialize_library()
def add():
    book_id=int(input("Enter Book ID: "))
    title=input("Enter Title: ").lower()
    author=input("Enter Author: ")
    price=int(input("Enter Price: "))
    library.add_book(Book(book_id,title,author,price))
    print("Book Added Successfully")

def remove_by_id():
    book_id=int(input("Enter Book ID: "))
    if library.remove_book_by_id(book_id):
        print("Book Removed Successfully")
    else:
        print("Book not found")

def remove_by_title():
    title=input("Enter Title: ").lower()
    if library.remove_book_by_title(title):
        print("Book Removed Successfully")
    else:
        print("Book not found")

def search_by_id():
    book_id=int(input("Enter Book ID: "))
    book=library.search_book_by_id(book_id)
    if book:
        print(book)
    else:
        print("Book not found")

def search_by_title():
    title=input("Enter Title: ").lower()
    book=library.search_book_by_title(title)
    if book:
        print(book)
    else:
        print("Book not found")
```


PYTHON
ClassPractice
Worksheet
AmicableNumber.py
LibraryManagementSystem.py
PermutationCombination.py
Student.py
TwinPrime.py
UserDictonary.py
LibraryManagementSystem.py > Library > add_book

```python
# ------------------------------------------------------------------------
while True:
    print("1. Add Book")
    print("2. Remove Book by ID")
    print("3. Remove Book by Title")
    print("4. Search Book by ID")
    print("5. Search Book by Title")
    print("6. List Books")
    print("7. No of Books in Library")
    print("8. Exit")
    choice=int(input("Enter your choice: "))
    if choice==1:
        add()
    elif choice==2:
        remove_by_id()
    elif choice==3:
        remove_by_title()
    elif choice==4:
        search_by_id()
    elif choice==5:
        search_by_title()
    elif choice==6:
        library.list_books()
    elif choice==7:
        print(f"No of Books in Library: {library.no_of_books_in_library()}")
    elif choice==8:
        break
    else:
        print("Invalid Choice")

    print("Want to continue? (y/n): ",end="")
    choice=input().lower()
    if choice != "y":
        break
```

Output (screenshots)



```
PS D:\Gaurav\MCA\Sem-1\Python> .\LibraryManagementSystem.py
1. Add Book
2. Remove Book by ID
3. Remove Book by Title
4. Search Book by ID
5. Search Book by Title
6. List Books
7. No of Books in Library
8. Exit
Enter your choice: 6
Book ID: 1 Title: Cracking the Coding Interview Author: gayle laakmann mcdowell Price: 300
Book ID: 2 Title: Data Structures and Algorithms Made Easy Author: narasimha karumanchi Price: 250
Book ID: 3 Title: Introduction to Algorithms Author: thomas h. cormen Price: 500
Book ID: 4 Title: The C Programming Language Author: brian w. kernighan Price: 150
Book ID: 5 Title: The C++ Programming Language Author: bjarne stroustrup Price: 200
Book ID: 6 Title: The Art of Computer Programming Author: donald knuth Price: 1000
Book ID: 7 Title: Python Crash Course Author: eric matthes Price: 200
Book ID: 8 Title: python programming: an introduction to computer science Author: John Zelle Price: 300
Book ID: 9 Title: Java: A Beginner's Guide Author: herbert schildt Price: 400
Book ID: 10 Title: Java: The Complete Reference Author: herbert schildt Price: 500
Book ID: 11 Title: Head First Java Author: kathy sierra Price: 600
Want to continue? (y/n): y
1. Add Book
2. Remove Book by ID
3. Remove Book by Title
4. Search Book by ID
5. Search Book by Title
6. List Books
7. No of Books in Library
8. Exit
Enter your choice: 7
No of Books in Library: 11
Want to continue? (y/n): n
PS D:\Gaurav\MCA\Sem-1\Python>
```

4) Learning outcomes (What I have learnt): Times new roman 12 size

1. Learn about classes and objects.

2. Learn about __init__() and __str__() function.

3. Learn about use of enumerator in loop

Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Quiz)		22
2.	Worksheet		8