Name: Fatima Bint e Naseer

Roll number: SU92-BSAIM-S24-050

Section: 2A

Semester: 2nd

Task = 09

**Program for managing different types of documents:**

**1. Define a parent class called Document with attributes title and author. Include a**

**method display\_info to display the title and author of the document.**

**2. Create a child class Book inheriting from Document. The Book class should have**

**additional attributes like genre and pages. Implement function overriding for the**

**display\_info method to include the genre and pages information.**

**3. Create another child class Article inheriting from Document. The Article class**

**should have additional attributes like journal and DOI (Digital Object Identifier).**

**Implement function overriding for the display\_info method to include the journal**

**and DOI information.**

**4. Implement function overloading in the Book class to handle different ways of**

**initializing a Book object. Allow initialization with just title and author, or title,**

**author, genre, and pages.**

**5. Implement function overloading in the Article class to handle different ways of**

**initializing Article object. Allow initialization with just title and author, or title,**

**author, journal, and DOI.**

**6. Implement file handling to store and retrieve information about books and articles.**

**Use text files to store the information in a structured format.**

class Document:

    def \_\_init\_\_(self, title, author):

        self.title = title

        self.author = author

    def display\_info(self):

        print(f"Title: {self.title}")

        print(f"Author: {self.author}")

class Book(Document):

    def \_\_init\_\_(self, title, author, genre=None, pages=None):

        super().\_\_init\_\_(title, author)

        self.genre = genre

        self.pages = pages

    def display\_info(self):

        super().display\_info()

        if self.genre:

            print(f"Genre: {self.genre}")

        if self.pages:

            print(f"Pages: {self.pages}")

    def initialize(self, title, author, genre=None, pages=None):

        self.title = title

        self.author = author

        self.genre = genre

        self.pages = pages

class Article(Document):

    def \_\_init\_\_(self, title, author, journal=None, doi=None):

        super().\_\_init\_\_(title, author)

        self.journal = journal

        self.doi = doi

    def display\_info(self):

        super().display\_info()

        if self.journal:

            print(f"Journal: {self.journal}")

        if self.doi:

            print(f"DOI: {self.doi}")

    def initialize(self, title, author, journal=None, doi=None):

        self.title = title

        self.author = author

        self.journal = journal

        self.doi = doi

def save\_documents\_to\_file(filename, documents):

    with open(filename, 'w') as file:

        for doc in documents:

            if isinstance(doc, Book):

                file.write(f"Book,{doc.title},{doc.author},{doc.genre},{doc.pages}\n")

            elif isinstance(doc, Article):

                file.write(f"Article,{doc.title},{doc.author},{doc.journal},{doc.doi}\n")

def read\_documents\_from\_file(filename):

    documents = []

    try:

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

            for line in file:

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

                if parts[0] == "Book":

                    documents.append(Book(parts[1], parts[2], parts[3], int(parts[4])))

                elif parts[0] == "Article":

                    documents.append(Article(parts[1], parts[2], parts[3], parts[4]))

    except FileNotFoundError:

        print("File not found. Starting with an empty document list.")

    return documents

filename = 'documents.txt'

documents = read\_documents\_from\_file(filename)

while True:

    print("\nDocument Management System")

    print("1. Add New Book")

    print("2. Add New Article")

    print("3. Display All Documents")

    print("4. Save Documents to File")

    print("5. Exit")

    choice = input("Enter your choice: ")

    if choice == '1':

        title = input("Enter book title: ")

        author = input("Enter author: ")

        genre = input("Enter genre: ")

        pages = int(input("Enter number of pages: "))

        new\_book = Book(title, author, genre, pages)

        documents.append(new\_book)

        print("Book added successfully.")

    elif choice == '2':

        title = input("Enter article title: ")

        author = input("Enter author: ")

        journal = input("Enter journal: ")

        doi = input("Enter DOI: ")

        new\_article = Article(title, author, journal, doi)

        documents.append(new\_article)

        print("Article added successfully.")

    elif choice == '3':

        print("\nDocument List:")

        for doc in documents:

            doc.display\_info()

            print()

    elif choice == '4':

        save\_documents\_to\_file(filename, documents)

        print("Documents saved to file successfully.")

    elif choice == '5':

        print("Exiting the program.")

        break

    else:

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