**Name: Usman Ul haq**

**Roll number: SU92-BSAIM-S24-035**

**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 DocumentBase:

    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 BookRecord(DocumentBase):

    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 and self.pages:

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

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

class ArticleRecord(DocumentBase):

    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 and self.doi:

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

            print(f"DOI: {self.doi}")

BOOKS\_FILE = "books.txt"

ARTICLES\_FILE = "articles.txt"

def save\_to\_file(record, file\_name):

    with open(file\_name, "a") as file:

        if isinstance(record, BookRecord):

            file.write(

                f"Book|{record.title}|{record.author}|{record.genre}|{record.pages}\n"

            )

        elif isinstance(record, ArticleRecord):

            file.write(

                f"Article|{record.title}|{record.author}|{record.journal}|{record.doi}\n"

            )

    print(f"{record.title} saved to {file\_name}")

def read\_from\_file(file\_name):

    try:

        with open(file\_name, "r") as file:

            for line in file:

                print(line.strip())

    except FileNotFoundError:

        print(f"No file found named '{file\_name}'.")

while True:

    print("1. Add a Book Record")

    print("2. Add an Article Record")

    print("3. Display Book Records")

    print("4. Display Article Records")

    print("5. Exit")

    choice = input("Enter your choice (1-5): ").strip()

    if choice == "1":

        title = input("Enter book title: ").strip()

        author = input("Enter book author: ").strip()

        genre = input("Enter book genre (optional): ").strip()

        pages = input("Enter number of pages (optional): ").strip()

        genre = genre if genre else None

        pages = int(pages) if pages.isdigit() else None

        book = BookRecord(title, author, genre, pages)

        save\_to\_file(book, BOOKS\_FILE)

    elif choice == "2":

        title = input("Enter article title: ").strip()

        author = input("Enter article author: ").strip()

        journal = input("Enter journal name (optional): ").strip()

        doi = input("Enter DOI (optional): ").strip()

        journal = journal if journal else None

        doi = doi if doi else None

        article = ArticleRecord(title, author, journal, doi)

        save\_to\_file(article, ARTICLES\_FILE)

    elif choice == "3":

        read\_from\_file(BOOKS\_FILE)

    elif choice == "4":

        read\_from\_file(ARTICLES\_FILE)

    elif choice == "5":

        break

    else:

        print("Invalid choice. Please enter a number between 1 and 5.")