Submitted By	**Habib ur Rehman (116)**
Subject	OOP
Assignment	Class Assignment 02
Date	Oct 09th , 2024

Submitted to:

Moderator	**Ms, Sajida Kalsoom**

ASSIGNMENT 02

"Mr. Books" is an online bookstore for all ages and interests. Mr. Books offers a wide range of books, from bestsellers to rare collector's items. The bookstore also provides services such as book recommendations, customer reviews, and personalized reading lists. As the lead software architect at Mr. Books, you are tasked to create a class diagram for the online system by keeping in view the following requirements: Books: There are different categories of books, such as FictionBook, NonFictionBook, and ScienceFictionBook. Each book has a title, author(s), ISBN, and price. Customers: Mr. Books customers can have a customer ID, name, and address. Reviews: Customers can leave reviews for books. Each review has a rating and comments. Recommendations: The system provides personalized book recommendations to customers based on their reading history. Orders: Customers can place orders for books. The system should be able to store order ID, order date, and the list of books ordered.

```java
import java.util.ArrayList;

import java.util.List;

class books{


    protected String title;

    protected String author;

    protected String ISBN;

    protected double price;


    books(){


    }
    books(String t,String A,String isbn,double p){

        this.title=t;

        this.author=A;

        this.ISBN=isbn;

        this.price=p;

    }
```

```java
void setTitle(String T){
    this.title=T;

}
void setAuthor(String a){
    this.author=a;

}
void setISBN(String isbn){
    this.ISBN=isbn;

}
void setPrice(double p){
    this.price=p;

}


String getTitle(){
    return title;
}
String getAuthor(){
    return author;
}
String getISBN(){
    return ISBN;
}
double getPrice(){
```

```java
        return price;

   }


   public String toString(){

      return "Title of the Book is: "+title+"\nits Author Name is: "+author+"\nISBN Number is:
"+ISBN+"\nPrice is: "+price;

   }


}

class bookCategories extends books{

   protected String bookType;

   bookCategories(){

   }

   bookCategories(String t,String author,String ISBN, double price,String type){
      super(t,author, ISBN, price);
      this.bookType=type;

   }

   public String toString(){

      return super.toString()+"Book Type is: "+bookType;
   }
```

```java
}


class customers{
    protected String name;
    protected String ID;
    protected String address;
    protected books book;
    customers(){

    }
customers(String N, String ID,String address,books b){

    this.name=N;
    this.ID=ID;
    this.address=address;
    this.book=b;
}

void setName(String Name){
    this.name=Name;

}
void setID(String ID){
    this.ID=ID;

}
void setAddress(String address){
```

```java
        this.address=address;

    }


    void setbooks(books b){
        this.book=b;
    }


    String getName(){
        return name;
    }


    String getID(){
        return ID;
    }
    String getaddress(){
        return address;
    }


    books getbooks(){
        return book;
    }


    public String toString(){
        return "Name of Customer is:"+name+"\n ID of the Customer is:"+ID+"\n The address of the
Customer is: "+address+"\n The book is: "+book;

    }


}
```

```java
class reviews extends customers{

    protected int rating;

    protected String comment;


    reviews(String N, String ID,String address,books book,String comment,int rating){

        super(N,ID,address,book);

        this.rating=rating;

        this.comment=comment;




    }


    void setRating(int r){

        this.rating=r;

    }

    void setComment(String c){

        this.comment=c;

    }


    String getComment(){

        return comment;

    }

    int getrating(){

        return rating;

    }

    // }
```

```java
    public String toString(){

        return super.toString()+"\nRating is: "+rating+"\n comment is:"+comment;


    }


}


class orders extends customers {
    protected String orderId;
    protected String date;
    List<books> bookList = new ArrayList<>();


    orders() {}


    orders(String N, String ID, String address, String id, String date, List<books> bookordered,
books b) {
        super(N, ID, address, b);
        this.orderId = id;
        this.date = date;
        bookList = bookordered;
    }


    void setOrderId(String id) {
        this.orderId = id;
    }


    void setDate(String date) {
```

```java
        this.date = date;

    }


    void setBooksOrdered(List<books> books) {

        this.bookList = books;

    }


    String getOrderId() {

        return orderId;

    }


    String getDate() {

        return date;

    }


    List<books> getBooksOrdered() {

        return bookList;

    }


    public String toString() {

        return super.toString() + "\nOrder ID: " + orderId + "\nDate: " + date + "\nBooks Ordered: "
+ bookList;

    }
}


class recommendation {

    private orders order;


    recommendation() {}
```

```java
    recommendation(orders order) {
      this.order = order;
    }


    public void setOrder(orders o) {
      this.order = o;
    }


    public books getRecommendation() {


      if (order != null && !order.getBooksOrdered().isEmpty()) {
        return order.getBooksOrdered().get(0);
      }
      return null; // If no books are found in order history
    }


    public String toString() {
      books recommendedBook = getRecommendation();
      return "Recommended book for you: " + (recommendedBook != null ? recommendedBook :
"No recommendation available");
    }
}


public class BookStore {
  public static void main(String[] args) {


    books book1 = new books("The great book", "F. Scott Fitzgerald", "123456789", 20.99);
```

```java
        books book2 = new books("Sapiens", "Yuval Noah Harari", "987654321", 25.99);
]

        List<books> orderedBooks = new ArrayList<>();

        orderedBooks.add(book1);

        orderedBooks.add(book2);


        orders order = new orders("habib", "C123", "123 Main St", "O1001", "2024-10-13",
orderedBooks, book1);

        System.out.println(order);


        // Creating a recommendation based on the previous order

        recommendation rec = new recommendation(order);

        System.out.println(rec);
    }


}
```

Class Diagram

Books

Title

Author

ISBN

Price

▲ ▲ ▲

| | |

| | |

+-------------+ +--------------+ +-----------------+

| FictionBook || NonFictionBook|| ScienceFictionBook|

Customer

ID

Name

address

orders

ID

Date

Book oredered

Reviews

Rating

Comment

book

Recommendation

Orders order

get recommendation
ordered history