Assignment Week 1 Java OOPS Concepts

Program 1: Imagine a publishing company which does not marketing for book and audio cassette versions.

Create a Class Publication that stores the title(a String) and price(type float) of publications. From this class derive two classes: Book which adds a page count(type Int) and Tap which adds a playing time in mins(Float). if an exception is caught, replace all the data member values with zero Value.

Book.java-

```
class Book extends Publication{
    int pageCount;



    public Book(String title, float price, int pageCount) {
        super(title, price);
        this.pageCount = pageCount;
    }
}
```

Tape.java-

```
class Tape extends Publication {
    float playingTime;

    public Tape(String title, float price, float playingTime) {
        super(title, price);
        this.playingTime = playingTime;
    }
}
```

Publication.java-

```
public class Publication {
    String title;
    float price;

    public Publication(String title, float price) {
        this.title = title;
        this.price = price;
    }
}
```

PublicationService.java-

```
import java.util.Scanner;
public class PublicationService {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Book book = null;
        Tape tape = null;

        try {

            System.out.println("Enter Book details:");
            System.out.print("Title: ");
            String bookTitle = scanner.nextLine();

            System.out.print("Price: ");
            float bookPrice = scanner.nextFloat();
            scanner.nextLine();

            System.out.print("Page Count: ");
            int pageCount = scanner.nextInt();

            book = new Book(bookTitle, bookPrice, pageCount);

            System.out.println("\nEnter Tape details:");
            System.out.print("Title: ");
            String tapeTitle = scanner.nextLine();
            // Consume the newline character left in the buffer
            scanner.nextLine();

            System.out.print("Price: ");
            float tapePrice = scanner.nextFloat();

            System.out.print("Playing Time (in mins): ");
            float playingTime = scanner.nextFloat();

            tape = new Tape(tapeTitle, tapePrice, playingTime);

        } catch (Exception e) {
            System.out.println("An exception occurred. Resetting data
members to zero values.");
            resetToZeroValues(book, tape);
        } finally {
            if (book != null) {
                System.out.println("\nBook Details:");
                displayBookDetails(book);
            }
```

```
            if (tape != null) {
                System.out.println("\nTape Details:");
                displaTapeDetails(tape);
            }

            scanner.close();
        }
    }

    private static void displayBookDetails(Book book) {
        System.out.println("Title: " + book.title);
        System.out.println("Price: " + book.price);
        System.out.println("Page Count: " + book.pageCount);
    }

    private static void displaTapeDetails(Tape tape) {
        System.out.println("Title: " + tape.title);
        System.out.println("Price: " + tape.price);
        System.out.println("Playing Time: " + tape.playingTime);
    }

    private static void resetToZeroValues(Book book, Tape tape) {
        System.out.println("Resetting data members to zero values.");

        if (book != null) {
            book.title = "";
            book.price = 0.0f;
            book.pageCount = 0;
        }

        if (tape != null) {
            tape.title = "";
            tape.price = 0.0f;
            tape.playingTime = 0.0f;
        }
    }
}
```

Output-

```
Enter Book details:
Title: Book 1
Price: 2500.00
Page Count: 456

Enter Tape details:
Title: Tape 1
Price: 45.60
Playing Time (in mins): 10

Book Details:
Title: Book 1
Price: 2500.0
Page Count: 456

Tape Details:
Title:
Price: 45.6
Playing Time: 10.0
```

Program 2: Design and implement the Library System.

Department.java-

```
public class Department {
    private String name;

    public Department(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    @Override
    public String toString() {
        return "Department [name=" + name + "]";
    }



}
```

Publisher.java-

```java
public class Publisher {
    private String name;
    private String contactInfo;

    public Publisher(String name, String contactInfo) {
        this.name = name;
        this.contactInfo = contactInfo;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getContactInfo() {
        return contactInfo;
    }

    public void setContactInfo(String contactInfo) {
        this.contactInfo = contactInfo;
    }

    @Override
    public String toString() {
        return "Publisher [name=" + name + ", contactInfo=" +
contactInfo + "]";
    }


}
```

LibraryBook.java-

```java
public class LibraryBook {
    private String name;
    private int pageCount;
    private double price;
    private int quantity;
    private Publisher publication;
    private String edition;
```

```
    private Department department;

    public LibraryBook(String name, int pageCount, double price, int quantity,
Publisher publication, String edition, Department department) {
        this.name = name;
        this.pageCount = pageCount;
        this.price = price;
        this.quantity = quantity;
        this.publication = publication;
        this.edition = edition;
        this.department = department;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public int getPageCount() {
        return pageCount;
    }

    public void setPageCount(int pageCount) {
        this.pageCount = pageCount;
    }

    public double getPrice() {
        return price;
    }

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

    public int getQuantity() {
        return quantity;
    }

    public void setQuantity(int quantity) {
        this.quantity = quantity;
    }

    public Publisher getPublication() {
        return publication;
    }

    public void setPublication(Publisher publication) {
        this.publication = publication;
```

```
    }

    public String getEdition() {
        return edition;
    }

    public void setEdition(String edition) {
        this.edition = edition;
    }

    public Department getDepartment() {
        return department;
    }

    public void setDepartment(Department department) {
        this.department = department;
    }

    @Override
    public String toString() {
        return "LibraryBook [name=" + name + ", pageCount=" + pageCount + ",
price=" + price + ", quantity=" + quantity
                + ", publication=" + publication + ", edition=" + edition + ",
department=" + department + "]";
    }


}
```

IssueRecord.java-

```
public class IssueRecord {
    private LibraryBook book;
    private String issuerId;
    private String issueDate;
    private String lastReturnDate;
    private double fine;

    public IssueRecord(LibraryBook book, String issuerId, String issueDate, String
lastReturnDate, double fine) {
        this.book = book;
        this.issuerId = issuerId;
        this.issueDate = issueDate;
        this.lastReturnDate = lastReturnDate;
        this.fine = fine;
    }

    public LibraryBook getBook() {
        return book;
    }
```

```
    public void setBook(LibraryBook book) {
        this.book = book;
    }

    public String getIssuerId() {
        return issuerId;
    }

    public void setIssuerId(String issuerId) {
        this.issuerId = issuerId;
    }

    public String getIssueDate() {
        return issueDate;
    }

    public void setIssueDate(String issueDate) {
        this.issueDate = issueDate;
    }

    public String getLastReturnDate() {
        return lastReturnDate;
    }

    public void setLastReturnDate(String lastReturnDate) {
        this.lastReturnDate = lastReturnDate;
    }

    public double getFine() {
        return fine;
    }

    public void setFine(double fine) {
        this.fine = fine;
    }

    @Override
    public String toString() {
        return "IssueRecord [book=" + book + ", issuerId=" + issuerId + ",
issueDate=" + issueDate + ", lastReturnDate="
                + lastReturnDate + ", fine=" + fine + "]";
    }

}
```

LibraryManagementSystem.java-

```
import java.util.ArrayList;
import java.util.Scanner;

public class LibraryManagementSystem {
```

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    ArrayList<Department> departments = new ArrayList<>();
    ArrayList<Publisher> publishers = new ArrayList<>();
    ArrayList<LibraryBook> books = new ArrayList<>();
    ArrayList<IssueRecord> issueRecords = new ArrayList<>();

    int choice;
    do {
        System.out.println("\nLibrary Management System Menu:");
        System.out.println("1. Add Department");
        System.out.println("2. Add Publisher");
        System.out.println("3. Add Book");
        System.out.println("4. Display Books");
        System.out.println("5. Issue Book");
        System.out.println("6. Display Issue Records");
        System.out.println("0. Exit");

        System.out.print("Enter your choice: ");
        choice = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        switch (choice) {
            case 1:
                System.out.print("Enter Department Name: ");
                String departmentName = scanner.nextLine();
                departments.add(new Department(departmentName));
                break;
            case 2:
                System.out.print("Enter Publisher Name: ");
                String publisherName = scanner.nextLine();
                System.out.print("Enter Publisher Contact Info: ");
                String contactInfo = scanner.nextLine();
                publishers.add(new Publisher(publisherName, contactInfo));
                break;
            case 3:
                System.out.print("Enter Book Name: ");
                String bookName = scanner.nextLine();
                System.out.print("Enter Page Count: ");
                int pageCount = scanner.nextInt();
                System.out.print("Enter Price: ");
                double price = scanner.nextDouble();
                System.out.print("Enter Quantity: ");
                int quantity = scanner.nextInt();
                scanner.nextLine(); // Consume the newline character
                System.out.print("Enter Edition: ");
                String edition = scanner.nextLine();
                System.out.print("Enter Department Name: ");
                String bookDepartment = scanner.nextLine();
```

```
                    Department department = findDepartment(departments,
bookDepartment);
                    System.out.print("Enter Publisher Name: ");
                    String bookPublisher = scanner.nextLine();
                    Publisher publisher = findPublisher(publishers, bookPublisher);

                    if (department != null && publisher != null) {
                        books.add(
                                new LibraryBook(bookName, pageCount, price,
quantity, publisher, edition, department));
                    } else {
                        System.out.println("Invalid Department or Publisher. Book
not added.");
                    }
                    break;
                case 4:
                    displayBooks(books);
                    break;
                case 5:
                    System.out.print("Enter Book Name to Issue: ");
                    String issueBookName = scanner.nextLine();
                    LibraryBook issueBook = findBook(books, issueBookName);

                    if (issueBook != null) {
                        System.out.print("Enter Issuer ID: ");
                        String issuerId = scanner.nextLine();
                        System.out.print("Enter Issue Date: ");
                        String issueDate = scanner.nextLine();
                        System.out.print("Enter Last Return Date: ");
                        String lastReturnDate = scanner.nextLine();
                        System.out.print("Enter Fine (if any): ");
                        double fine = scanner.nextDouble();

                        issueRecords.add(new IssueRecord(issueBook, issuerId,
issueDate, lastReturnDate, fine));
                        System.out.println("Book issued successfully.");
                    } else {
                        System.out.println("Book not found. Cannot issue.");
                    }
                    break;
                case 6:
                    displayIssueRecords(issueRecords);
                    break;
                case 0:
                    System.out.println("Exiting Library Management System.");
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }

        } while (choice != 0);
```

```
        scanner.close();
    }

    private static void displayBooks(ArrayList<LibraryBook> books) {
        System.out.println("\nLibrary Books:");
        for (LibraryBook book : books) {
            System.out.println(book.toString());
        }
    }

    private static void displayIssueRecords(ArrayList<IssueRecord> issueRecords) {
        System.out.println("\nIssue Records:");
        for (IssueRecord record : issueRecords) {
            System.out.println(record.toString());
        }
    }

    private static Department findDepartment(ArrayList<Department> departments,
String departmentName) {
        for (Department department : departments) {
            if (department.getName().equalsIgnoreCase(departmentName)) {
                return department;
            }
        }
        System.out.println("Department not found.");
        return null;
    }

    private static Publisher findPublisher(ArrayList<Publisher> publishers, String
publisherName) {
        for (Publisher publisher : publishers) {
            if (publisher.getName().equalsIgnoreCase(publisherName)) {
                return publisher;
            }
        }
        System.out.println("Publisher not found.");
        return null;
    }

    private static LibraryBook findBook(ArrayList<LibraryBook> books, String
bookName) {
        for (LibraryBook book : books) {
            if (book.getName().equalsIgnoreCase(bookName)) {
                return book;
            }
        }
        System.out.println("Book not found.");
        return null;
    }
}
```

Output-

```
Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 1
Enter Department Name: CSE

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 1
Enter Department Name: ECE

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 2
Enter Publisher Name: Publisher 1
Enter Publisher Contact Info: Address 1

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 2
Enter Publisher Name: Publisher 2
```

Enter Publisher Contact Info: Address 2

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 3
Enter Book Name: Book 1
Enter Page Count: 200
Enter Price: 1000
Enter Quantity: 20
Enter Edition: 2020
Enter Department Name: CSE
Enter Publisher Name: Publisher 1

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 4

Library Books:
LibraryBook [name=Book 1, pageCount=200, price=1000.0, quantity=20, publication=Publisher [name=Publisher 1, contactInfo=Address 1], edition=2020, department=Department [name=CSE]]

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 5
Enter Book Name to Issue: Book 1
Enter Issuer ID: 20256
Enter Issue Date: 20/02/2023
Enter Last Return Date: 20/03/2023

Enter Fine (if any): 0
Book issued successfully.

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 6

Issue Records:
IssueRecord [book=LibraryBook [name=Book 1, pageCount=200, price=1000.0, quantity=20, publication=Publisher [name=Publisher 1, contactInfo=Address 1], edition=2020, department=Department [name=CSE]], issuerId=20256, issueDate=20/02/2023, lastReturnDate=20/03/2023, fine=0.0]

Library Management System Menu:
1. Add Department
2. Add Publisher
3. Add Book
4. Display Books
5. Issue Book
6. Display Issue Records
0. Exit
Enter your choice: 0
Exiting Library Management System.