GUVI-Task_3

1.

1.Design a Java program that uses OOP principles to model the Book Create two classes: Book and Library. The Book class should have attributes such as bookID, title, author, and is Available.
The Library class should include an array to store book objects.
2.Provide methods to add books, remove book search books (using id)and display books.Write a Java program that demonstrates the use of these classes and methods by allowing the user to interact with the library system.

```
package Task_3.Solution_1;


public class Book {
   private int bookID;
   private String title;
   private String author;
   private boolean isAvailable;
   public Book(int bookID,String title,String author,boolean isAvailable){
      this.bookID=bookID;
      this.title=title;
      this.author=author;
      this.isAvailable=isAvailable;
   }
   public int getBookID() {
      return bookID;
   }
   public String getTitle() {
      return title;
   }
   public String getAuthor() {
      return author;
   }
   public boolean isAvailable() {
      return isAvailable;
   }
   public void display(){
      System.out.println("Book ID:"+ bookID);
      System.out.println("Title:"+ title);
```

```
        System.out.println("Author:"+ author);
        System.out.println("Availability:"+ isAvailable);
    }
}
```

```
package Task_3.Solution_1;

class Library {
    Book[] books;
    int count;

    Library(int size) {
        books = new Book[size];
        count = 0;
    }

    void addBook(Book b) {
        if (count < books.length) {
            books[count++] = b;
            System.out.println("Book added");
        }
    }
    void removeBook(int bookID) {
        boolean found = false;
        for (int i = 0; i < count; i++) {
            if (books[i].getBookID() == bookID) {
                books[i] = books[count - 1];
                books[count - 1] = null;
                count--;
                found = true;
                System.out.println("Book removed");
                break;
            }
        }
    }
    void searchBook(int bookID) {
        for (int i = 0; i < count; i++) {
            if (books[i].getBookID() == bookID) {
                System.out.println("Book found:");
                books[i].display();
                return;
```

```
        }
      }
    }

    void displayAllBooks() {
      for (int i = 0; i < count; i++) {
        books[i].display();
      }
    }
}
```

```
package Task_3.Solution_1;

import java.util.Scanner;

public class BookManagementSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Library lib = new Library(5);
        int choice;

        while (true) {
            System.out.println("1. Add Book");
            System.out.println("2. Remove Book");
            System.out.println("3. Search Book");
            System.out.println("4. Display All Books");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");
            choice = sc.nextInt();
            switch (choice) {
                case 1:
                    System.out.print("Enter Book ID: ");
                    int id = sc.nextInt();
                    System.out.print("Enter Title: ");
                    String title = sc.next();
                    System.out.print("Enter Author: ");
                    String author = sc.next();
                    System.out.print("Availablity (true/false): ");
                    boolean available = sc.nextBoolean();
                    lib.addBook(new Book(id, title, author, available));
                    break;
```

```
            case 2:
                System.out.print("Enter Book ID to remove: ");
                int removeId = sc.nextInt();
                lib.removeBook(removeId);
                break;
            case 3:
                System.out.print("Enter Book ID to search: ");
                int searchId = sc.nextInt();
                lib.searchBook(searchId);
                break;
            case 4:
                lib.displayAllBooks();
                break;
            case 5:
                System.out.println("Exiting ...");
                break;

            default:
                System.out.println("Invalid choice");
        }
    }
  }
}
```

Output:

1. Add Book
2. Remove Book
3. Search Book
4. Display All Books
5. Exit
Enter your choice: **1**
Enter Book ID: **1**
Enter Title: ABC
Enter Author: abc
Availablity (true/false): true
Book added
1. Add Book
2. Remove Book
3. Search Book
4. Display All Books
5. Exit

Enter your choice: **1**
Enter Book ID: **2**
Enter Title: DEF
Enter Author: def
Availablity (true/false): false
Book added
1. Add Book
2. Remove Book
3. Search Book
4. Display All Books
5. Exit
Enter your choice: **1**
Enter Book ID: **3**
Enter Title: GHI
Enter Author: ghi
Availablity (true/false): true
Book added
1. Add Book
2. Remove Book
3. Search Book
4. Display All Books
5. Exit
Enter your choice: **2**
Enter Book ID to remove: **3**
Book removed
1. Add Book
2. Remove Book
3. Search Book
4. Display All Books
5. Exit
Enter your choice: **3**
Enter Book ID to search: **2**
Book found:
Book ID:**2**
Title:DEF
Author:def
Availability:false
1. Add Book
2. Remove Book
3. Search Book
4. Display All Books

5. Exit
Enter your choice: **4**
Book ID:**1**
Title:ABC
Author:abc
Availability:true
Book ID:**2**
Title:DEF
Author:def
Availability:false
1. Add Book
2. Remove Book
3. Search Book
4. Display All Books
5. Exit
Enter your choice: **5**
Exiting ...

2.Create Interface Taxable with members sales Tax-7% and income Tax-10.5%, create abstract method calcTax().

a. Create class Employee(empId,name,salary) and implement Taxable to calculate income Tax on yearly salary.

b. Create class Product(pid.price, quantity) and implement Taxable to calculate sales Tax on unit price of product.

c. Create class for main method(Say DriverMain), accept employee information and a product information from user and print income tax and sales tax respectively

```
package Task_3.Solution_2;

class Employee implements Taxable {
    int empID;
    String name;
    double salary;

    Employee(int empID,String name,double salary) {
        this.empID =empID;
```

```
        this.name = name;
        this.salary =salary;
    }

    public void calcTax() {
        double tax = salary*12*incomeTax;
        System.out.println("Income Tax for "+name+": "+tax);
    }
}
```

```
package Task_3.Solution_2;

class Product implements Taxable {
    int pID;
    double price;
    int quantity;

    Product(int pID,double price,int quantity) {
        this.pID =pID;
        this.price =price;
        this.quantity =quantity;
    }

    public void calcTax() {
        double tax = price*salesTax;
        System.out.println("Sales Tax on one unit: "+tax);
    }
}
```

```
package Task_3.Solution_2;

interface Taxable {
    double salesTax = 0.07;     // 7/100
    double incomeTax = 0.105;   // 10.5/100
    void calcTax();
}
```

```
package Task_3.Solution_2;

import java.util.Scanner;
public class DriverMain {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      // To check income tax for the employee
      System.out.println("Enter Employee ID, Name, and Monthly Salary:");
      int empID = sc.nextInt();
      String name = sc.next();
      double salary = sc.nextDouble();
      Employee employee = new Employee(empID, name, salary);
      employee.calcTax();

      // To check sales tax for the product
      System.out.println("Enter Product ID, Price, and Quantity:");
      int pID = sc.nextInt();
      double price = sc.nextDouble();
      int quantity = sc.nextInt();
      Product product = new Product(pID, price, quantity);
      product.calcTax();
   }
}
```

Output:

```
Enter Employee ID, Name, and Monthly Salary:
1
Regin
56000.00
Income Tax for Regin: 70560.0
Enter Product ID, Price, and Quantity:
1
72.00
2
Sales Tax on one unit: 5.040000000000001
```