**JFS TASK-3**

**QUESTION 1**

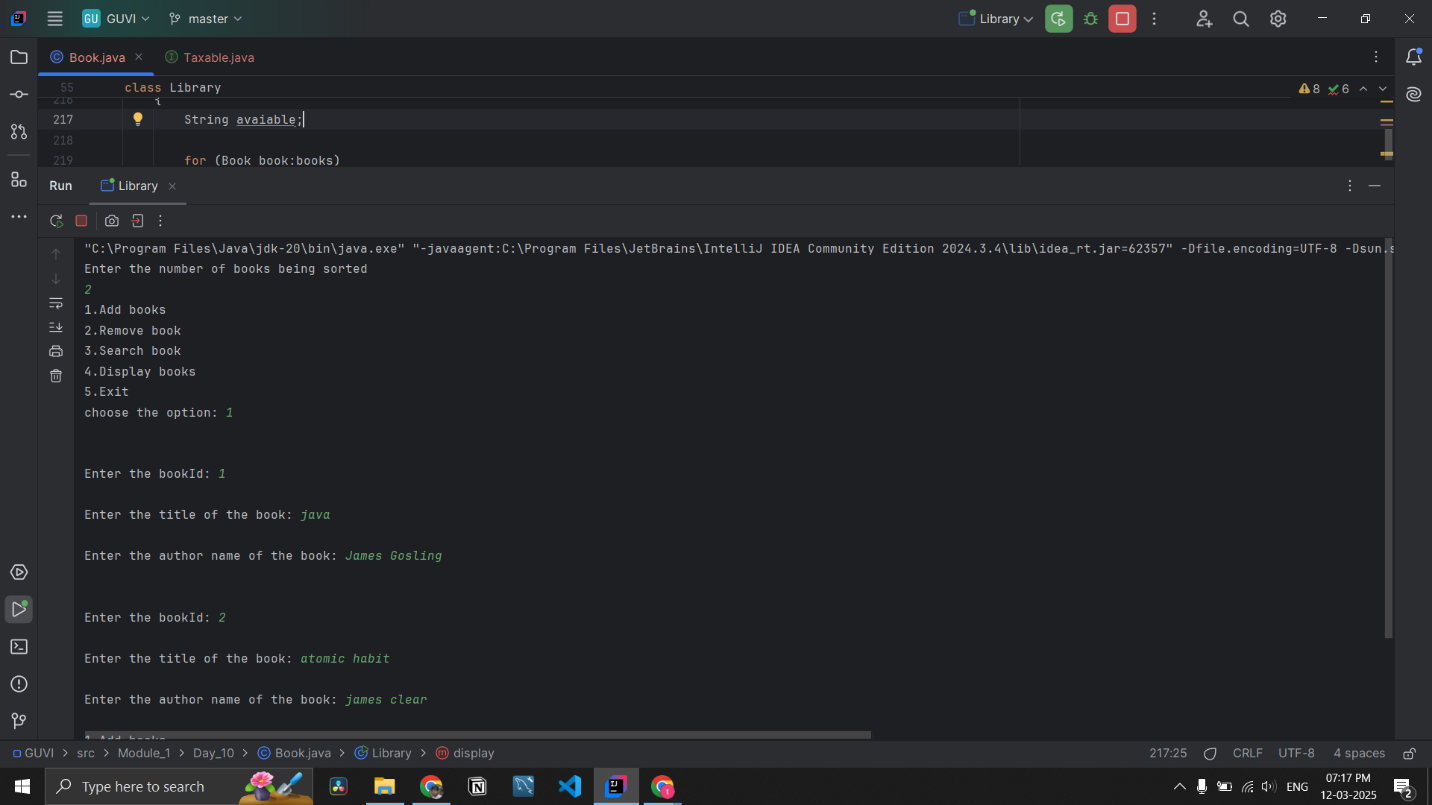
**Task Descriptions: Management System**

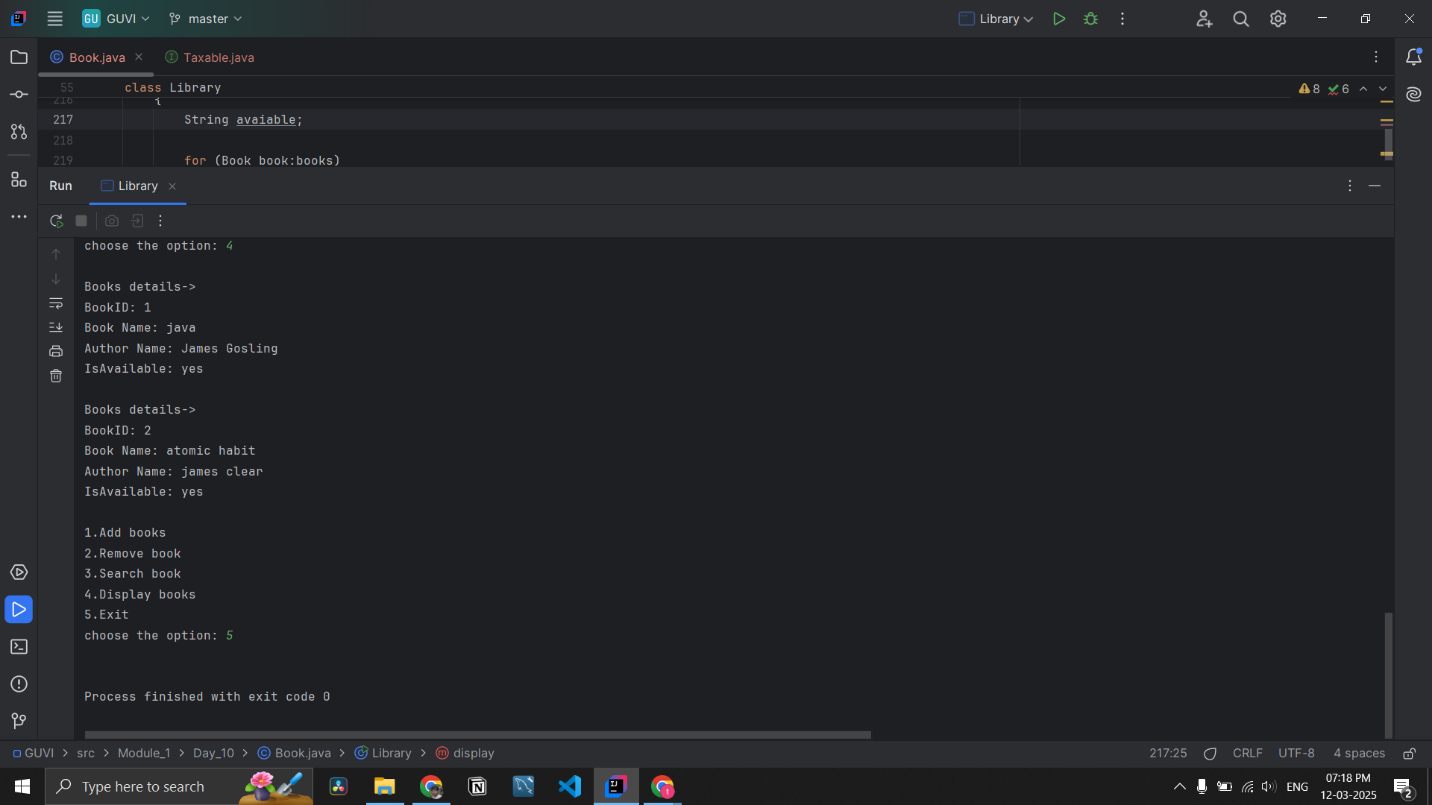
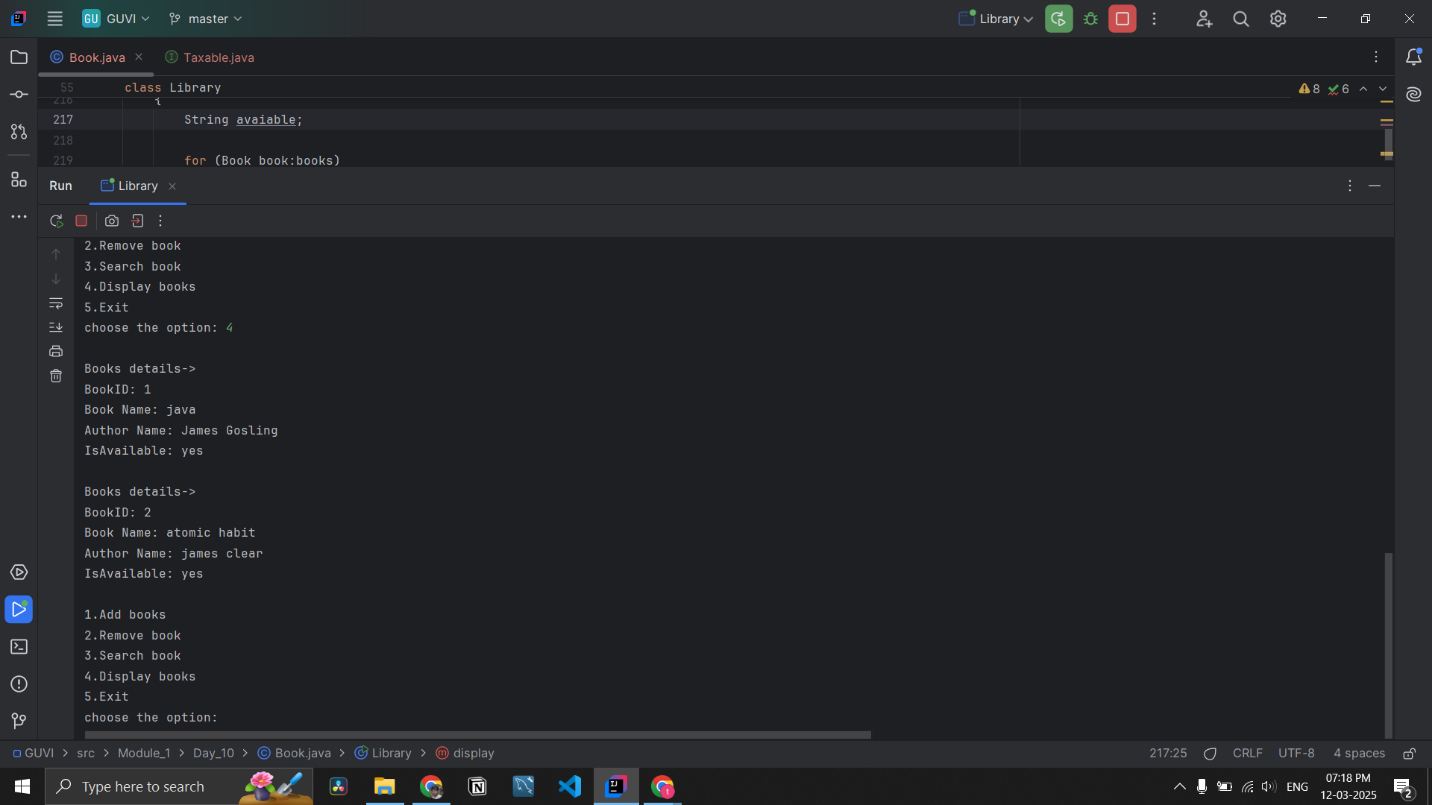
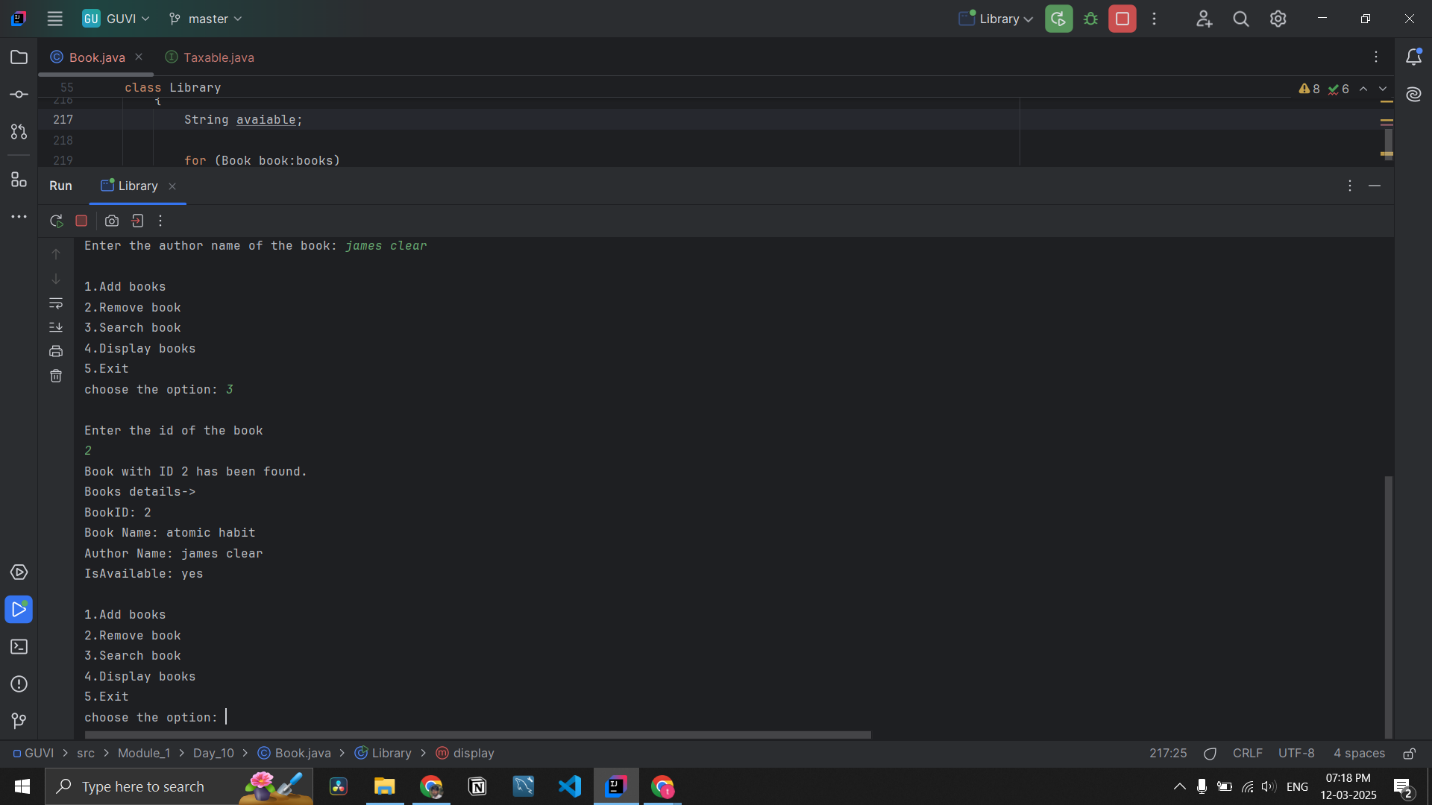
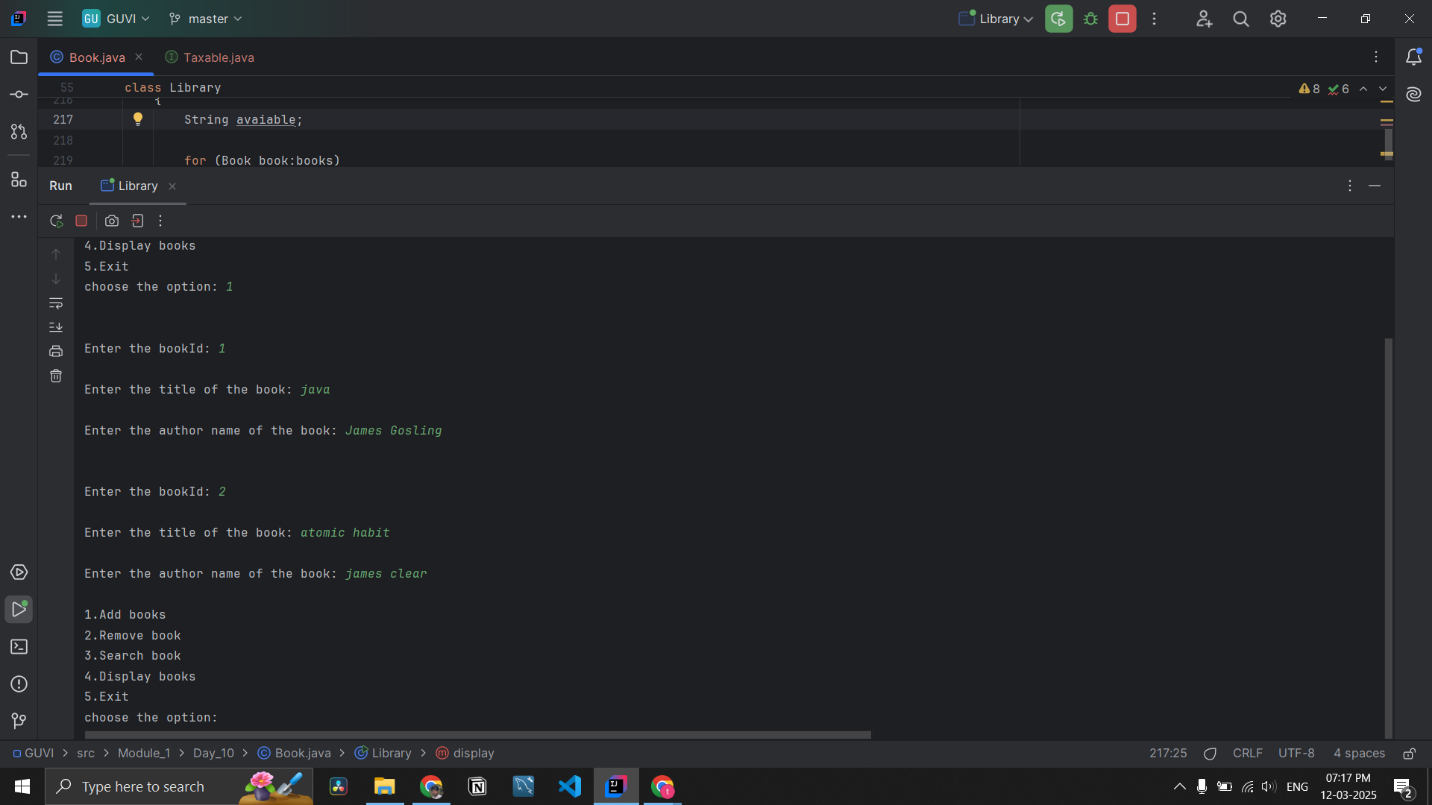
1. **Design a Java program that uses OOP principles to model the Book. Create two classes: Book and Library. The Book class should have attributes such as bookID, title, author, and isAvailable. The Library class should include an array to store book objects.**
2. **Provide methods to add books, remove books, search books (using ID), and display books.**

**Code:**

package Module\_1.Day\_10;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
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;  
 }  
  
 // Getter methods  
 public int getBookID()  
 {  
 return bookID;  
 }  
 public String getTitle()  
 {  
 return title;  
 }  
 public String getAuthor()  
 {  
 return author;  
 }  
 public boolean getIsAvailable()  
 {  
 return isAvailable;  
 }  
  
 //setter method  
 public void setBookID(int bookID)  
 {  
 this.bookID=bookID;  
 }  
 public void setTitle(String title)  
 {  
 this.title=title;  
 }  
 public void setAuthor(String author) {  
 this.author = author;  
 }  
 public void setIsAvailable(boolean isAvailable) {  
 this.isAvailable = isAvailable;  
 }  
}  
  
public class Library  
{  
  
 public static void main(String[] args) {  
 Scanner input=new Scanner(System.*in*);  
 Library library=new Library();  
// ArrayList<Book> books=new ArrayList<>();  
 System.*out*.println("Enter the number of books being sorted");  
 int size=input.nextInt();  
 Book [] book=new Book[size];  
  
 boolean flag=true;  
 int bookcount=0;  
 while (flag)  
 {  
 System.*out*.println(  
 "1.Add books \n" +  
 "2.Remove book\n" +  
 "3.Search book\n" +  
 "4.Display books\n" +  
 "5.Exit");  
  
 System.*out*.print("choose the option: ");  
  
 int choice=input.nextInt();  
 input.nextLine();  
  
 System.*out*.println();  
 switch (choice)  
 {  
 case 1:  
 {  
 if(bookcount<size) {  
  
 for (int i = 0; i < book.length; i++) {  
// Scanner input=new Scanner(System.in);  
 System.*out*.println();  
 System.*out*.print("Enter the bookId: ");  
 int bookid = Integer.*parseInt*(input.nextLine()); // storing book id  
 System.*out*.println();  
  
 System.*out*.print("Enter the title of the book: ");  
 String title = input.nextLine(); // Storing title of the book  
 System.*out*.println();  
  
 System.*out*.print("Enter the author name of the book: ");  
 String author = input.nextLine();  
 System.*out*.println();  
 //Storing author name  
 book[i] = new Book(bookid, title, author, true); // storing the data in an object in the book class  
 bookcount++;  
 }  
  
 }  
 else {  
 System.*out*.println("Library is full. Cannot add more books\n");  
 }  
  
 break;  
 }  
 case 2:  
 {  
 // removing a book from the class  
 System.*out*.print("Enter the id of the book: ");  
 int id=input.nextInt();  
 library.Removebook(book,id);  
 break;  
 }  
 case 3:  
 {  
 //searching the book was in the class or not  
 System.*out*.println("Enter the id of the book");  
 int id=input.nextInt();  
 library.Searchbook(book,id);  
 break;  
 }  
 case 4:  
 {  
 library.display(book);  
 break;  
 }  
  
 case 5:  
 {  
 //use to close the program  
 flag=false;  
 break;  
 }  
 default:  
 {  
 System.*out*.println("Invalid input try again");  
 }  
 }  
 }  
  
  
 }  
  
  
 public Book[] Removebook(Book[] books,int id)  
 {  
 boolean found = false; // using to book was present or not  
 int index\_to\_remove=-1; // use to store which book we're removing index values  
 for (int i = 0; i < books.length; i++)  
 {  
  
 if(books[i].getBookID()==id)  
 {  
 index\_to\_remove=i; //storing the index of the removing book  
 found=true;  
 break;  
 }  
  
 }  
  
 // removing the book from the array  
 Book[] copy\_book=new Book[books.length-1]; //we're copying the book full data in another new object  
  
 int index=0; // we're using this index to make to remove the book  
  
 for (int i = 0; i < copy\_book.length; i++)  
 {  
 // if the removing book index gets, we don't add the data in the new book  
 if (i!=index\_to\_remove)  
 {  
 copy\_book[index++]=books[i];  
 }  
  
 }  
 if(!found) {  
 System.*out*.println("Book with ID " + id + " not found.");  
  
 }  
 return copy\_book;  
 }  
  
 public void Searchbook(Book[] books,int id)  
 {  
 String avaiable;   
 for (int i = 0; i < books.length; i++)  
 {  
 if(books[i].getBookID()==id) //checking the book was present in the book class  
 {  
 if(books[i].getIsAvailable())  
 {  
 avaiable="yes";  
 }  
 else {  
 avaiable="No";  
 }  
 System.*out*.println("Book with ID " + id + " has been found.");  
 System.*out*.println("Books details-> ");  
 System.*out*.println("BookID: "+books[i].getBookID()+"\nBook Name: "+ books[i].getTitle()+"\nAuthor Name: "+books[i].getAuthor()+"\nIsAvailable: "+avaiable);  
 System.*out*.println();  
 break;  
 }  
  
 }  
  
 }  
  
 public void display(Book[] books)  
 {  
 String avaiable;  
  
 for (Book book:books)  
 {  
 if(book.getIsAvailable())  
 {  
 avaiable="yes";  
 }  
 else {  
 avaiable="No";  
 }  
 System.*out*.println("Books details-> ");  
 System.*out*.println("BookID: "+book.getBookID()+"\nBook Name: "+ book.getTitle()+"\nAuthor Name: "+book.getAuthor()+"\nIsAvailable: "+avaiable);  
 System.*out*.println();  
 }  
  
  
 }  
  
}

**output screenshots:**

****



**QUESTION 2**

**2. Create Interface Taxable with members salesTax=7% and incomeTax=10.5%. Create abstract method calcTax().  
a. Create class Employee(empId, name, salary) and implement Taxable to calculate incomeTax on yearly salary.  
b. Create class Product(pid, price, quantity) and implement Taxable to calculate salesTax 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.**

**code:**

package Module\_1.Day\_10;  
  
import java.util.Scanner;  
  
public interface Taxable  
{  
 double *salesTax* = 0.07;  
 double *incomeTax*=0.105;  
  
 public void calTax(); // rule to child class to implemented  
}  
  
class Employee implements Taxable  
{  
 int empId;  
 String name;  
 double salary;  
  
 public Employee(int empId, String name, double salary) {  
 this.empId = empId;  
 this.name = name;  
 this.salary = salary;  
 }  
  
 // default or empty construct to avoid error in objection creation without sent data time  
 public Employee() {  
  
 }  
  
 // calculating the income tax for the salary  
 @Override  
 public void calTax()  
 {  
 long income\_tax= (long) ((long)salary\**incomeTax*);  
 System.*out*.println("Income tax for a salary of " + salary + " is " + income\_tax + ".");  
 }  
}  
  
class products implements Taxable  
{  
 int pid;  
 double price;  
 int quantity;  
  
 public products(int pid, double price, int quantity) {  
 this.pid = pid;  
 this.price = price;  
 this.quantity = quantity;  
 }  
  
 // calculating the sales tax for the product  
 @Override  
 public void calTax()  
 {  
 double sales\_tax= price\**salesTax*;  
 System.*out*.println("Sales Tax on Product: " + sales\_tax +"Rupees");  
  
 }  
}  
  
class DriverMain  
{  
 public static void main(String[] args) {  
 Scanner input=new Scanner(System.*in*);  
  
 // accesses employee information  
 System.*out*.print("Enter your id :");  
 int id=input.nextInt(); //storing id information  
 input.nextLine();  
  
 System.*out*.print("Enter you name: ");  
  
 String name=input.nextLine();  
  
 System.*out*.print("Enter your yearly salary: ");  
  
 double salary=input.nextDouble();  
 input.nextLine();  
  
 Employee employee=new Employee(id,name,salary); // sending the data to employee class  
 employee.calTax(); //calling the function to calculate the tax  
  
  
 // accesses product information  
  
 System.*out*.print("Enter ProductId(Pid): ");  
 int pid=Integer.*parseInt*(input.nextLine());  
  
 System.*out*.print("Enter price of the Product: ");  
 long price=Long.*parseLong*(input.nextLine());  
  
 System.*out*.print("Enter the quantity of the product : ");  
 int quantity=Integer.*parseInt*(input.nextLine());  
  
 products products=new products(pid,price,quantity); // sending the data to products class  
  
 products.calTax(); //calling the function to calculate the tax  
 }  
  
}

**Output screenshot:**

