Code for 1)

//Class Book  
public class Book {  
 int bookid;  
 String title;  
 String authorname;  
 boolean isAvailable;  
//constructor with attributes  
 public Book(int bookid, String title, String authorname,boolean isAvailable) {  
 this.bookid = bookid;  
 this.title = title;  
 this.authorname = authorname;  
 this.isAvailable=isAvailable;  
 }  
  
 Book()  
 { }  
 public int getBookid() {  
 return bookid;  
 }  
  
 public String getTitle() {  
 return title;  
 }  
  
 public String getAuthorname() {  
 return authorname;  
 }  
  
 public boolean getisAvailable() {  
 return isAvailable;  
 }  
  
 public void setBookid(int bookid) {  
 this.bookid = bookid;  
 }  
  
 public void setTitle(String title) {  
 this.title = title;  
 }  
  
 public void setAuthorname(String authorname) {  
 this.authorname = authorname;  
 }  
  
 public boolean setAvailable(boolean available) {  
 this.isAvailable = available;  
  
 return available;  
 }  
}

import java.util.Scanner;  
//class Library  
public class Library extends Book {  
 public Book[] books = new Book[5];  
  
  
 public Library(int bookid, String title, String authorname,boolean isAvailable) {  
 super(bookid, title, authorname,isAvailable);  
  
 }  
  
 int i = 0;  
  
 public Library() {  
 super();  
 }  
  
 Scanner scanner = new Scanner(System.*in*);  
//method to add book to library system  
 public void addBook() {  
  
  
 System.*out*.println("Enter the book id :");  
  
  
 bookid = scanner.nextInt();  
 setBookid(bookid);  
  
 System.*out*.print("Enter book name: ");  
 title = scanner.next();  
 setTitle(title);  
  
 System.*out*.print("Enter author name: ");  
 authorname = scanner.next();  
 setAuthorname(authorname);  
 boolean isAvailable= setAvailable(true);  
  
 // Create a new Product object and store it in the array  
  
 books[i] = new Book(bookid, title, authorname,isAvailable);  
  
 i++;  
  
  
 }  
//method to display books in library  
 public void display(Book[] books) {  
  
 for (int j = 0; j < i; j++) {  
 System.*out*.println("BookID: "+books[j].getBookid());  
 System.*out*.println("Book name: "+books[j].getTitle());  
 System.*out*.println("Author name: "+books[j].getAuthorname());  
 System.*out*.println("book is Available: "+books[j].getisAvailable());  
  
  
 }  
 }  
//method to remove book with bookid  
 public void remove(int bookid) {  
 if (i == 0) System.*out*.println("no books to remove");  
 else {  
  
 for (int j = 0; j < i-1; j++) {  
  
 if (books[j].bookid == bookid) {  
 while (j < i-1) {  
 books[j] = books[j + 1];  
 j++;  
 }  
  
  
 }  
 }  
 this.i--;  
 }  
  
 }  
  
  
}

import java.util.Scanner;  
// class BookManagementSystem  
public class BookManagementSystem {  
 public static void main(String[] args) {  
 Library lib = new Library();  
 Scanner scanner = new Scanner(System.*in*);  
 int choice;  
 do {  
  
 System.*out*.println("1. Add Book");  
 System.*out*.println("2. Remove Book");  
 System.*out*.println("3. Display Books");  
 System.*out*.println("4. Exit");  
 System.*out*.print("Enter your choice: ");  
 choice = scanner.nextInt();  
 scanner.nextLine();  
 // Consume the newline character  
  
 switch (choice) {  
 case 1:  
  
 lib.addBook();  
 break;  
 case 2:  
 lib.display(lib.books);  
 System.*out*.println("Enter the book Id to remove");  
 int bookid = scanner.nextInt();  
  
 lib.remove(bookid);  
 break;  
 case 3:  
 lib.display(lib.books);  
 break;  
 case 4:  
 System.*out*.println("Exiting the Library Management System");  
 break;  
 default:  
 System.*out*.println("Invalid choice. Please try again.");  
 }}  
 while (choice != 4) ;  
  
 scanner.close();  
 }  
  
  
 }

Output:

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 1

Enter the book id :

101

Enter book name: Hamlet

Enter author name: Shakespeare

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 1

Enter the book id :

102

Enter book name: Beloved

Enter author name: ToniMorrison

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 1

Enter the book id :

103

Enter book name: WarAndPeace

Enter author name: LeoTolstoy

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 1

Enter the book id :

104

Enter book name: PrideAndPrejudice

Enter author name: JaneAusten

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 1

Enter the book id :

105

Enter book name: ToTheLighthouse

Enter author name: VirginiaWoolf

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 2

BookID: 101

Book name: Hamlet

Author name: Shakespeare

book is Available: true

BookID: 102

Book name: Beloved

Author name: ToniMorrison

book is Available: true

BookID: 103

Book name: WarAndPeace

Author name: LeoTolstoy

book is Available: true

BookID: 104

Book name: PrideAndPrejudice

Author name: JaneAusten

book is Available: true

BookID: 105

Book name: ToTheLighthouse

Author name: VirginiaWoolf

book is Available: true

Enter the book Id to remove

102

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 3

BookID: 101

Book name: Hamlet

Author name: Shakespeare

book is Available: true

BookID: 103

Book name: WarAndPeace

Author name: LeoTolstoy

book is Available: true

BookID: 104

Book name: PrideAndPrejudice

Author name: JaneAusten

book is Available: true

BookID: 105

Book name: ToTheLighthouse

Author name: VirginiaWoolf

book is Available: true

1. Add Book

2. Remove Book

3. Display Books

4. Exit

Enter your choice: 4

Exiting the Library Management System

Process finished with exit code 0

//////////////////////////////////////////////////////////////////////

Code for 2)

public interface Taxable {  
  
 double *salesTax* = 7.0;  
 double *incomeTax* = 10.5;  
  
 // Methods to calculate taxes  
 double calculateIncomeTax(double yearlySalary);  
 double calculateSalesTax(double unitPrice);  
}

//class Employee  
public abstract class Employee implements Taxable {  
  
 private int empID;  
 private String name;  
 private double salary;  
  
 // Constructor  
 public Employee(int empID, String name, double salary) {  
 this.empID = empID;  
 this.name = name;  
 this.salary = salary;  
 }  
  
 // Implementation of calculateIncomeTax  
 @Override  
 public double calculateIncomeTax(double yearlySalary) {  
 return yearlySalary \* (*incomeTax* / 100);  
 }  
  
 // Method to calculate and display income tax  
 public void displayIncomeTax() {  
 double yearlySalary = salary \* 12; // Convert monthly salary to yearly salary  
 double incomeTax = calculateIncomeTax(yearlySalary);  
 System.*out*.println("Yearly income Tax for Employee " + name + " with empID: " + empID + " is " + incomeTax);  
 }  
}

//Class Product  
public abstract class Product implements Taxable {  
  
 private int pid;  
 private double price;  
 private int quantity;  
  
 // Constructor  
 public Product(int pid, double price, int quantity) {  
 this.pid = pid;  
 this.price = price;  
 this.quantity = quantity;  
 }  
//implementation of calculateSalesTax  
 @Override  
 public double calculateSalesTax(double unitPrice) {  
 return unitPrice \* (*salesTax* / 100);  
 }  
  
 // Method to calculate and display sales tax  
 public void displaySalesTax() {  
 double unitprice = price/quantity;  
 double salesTax = calculateSalesTax(unitprice);  
 System.*out*.println("Sales Tax for Product ID " + pid + " " + salesTax + " per unit.");  
 }  
}

import java.util.Scanner;  
//Class DriverMain  
public class DriverMain {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
  
 // Accept Employee information  
 System.*out*.println("Enter Employee Details:");  
 System.*out*.print("Employee ID: ");  
 int empID = scanner.nextInt();  
 scanner.nextLine(); // Consume newline  
 System.*out*.print("Name: ");  
 String name = scanner.nextLine();  
 System.*out*.print("Monthly Salary: ");  
 double salary = scanner.nextDouble();  
  
 // Create Employee object and display income tax  
 Employee employee = new Employee(empID, name, salary) {  
 @Override  
 public double calculateSalesTax(double unitPrice) {  
 return 0;  
 }  
 };  
 employee.displayIncomeTax();  
  
 // Accept Product information  
 System.*out*.println("Enter Product Details:");  
 System.*out*.print("Product ID: ");  
 int pid = scanner.nextInt();  
 System.*out*.print("Total price: ");  
 double price = scanner.nextDouble();  
 System.*out*.print("Quantity: ");  
 int quantity = scanner.nextInt();  
  
 // Create Product object and display sales tax  
 Product product = new Product(pid, price, quantity) {  
 @Override  
 public double calculateIncomeTax(double yearlySalary) {  
 return 0;  
 }  
 };  
 product.displaySalesTax();  
  
  
 }  
}

Output:

Enter Employee Details:

Employee ID: 10001

Name: Balajyothi

Monthly Salary: 50000

Yearly income Tax for Employee Balajyothi with empID: 10001 is 63000.0

Enter Product Details:

Product ID: 501

Total price: 40000

Quantity: 40

Sales Tax for Product ID 501 70.0 per unit.

Process finished with exit code 0