Soluion-1:

```
package task3;

import java.util.Scanner;

// Book class to represent a book in the library
class Book {
  // Attributes of the Book class
  private int bookID;
  private String title;
  private String author;
  private boolean isAvailable;

  // Constructor for the Book class
  public Book(int bookID, String title, String author, boolean isAvailable)
{
    this.bookID = bookID;
    this.title = title;
    this.author = author;
    this.isAvailable = isAvailable;
  }

  // Getters for the attributes
  public int getBookID() {
    return bookID;
  }

  public String getTitle() {
    return title;
  }

  public String getAuthor() {
    return author;
  }

  public boolean isAvailable() {
    return isAvailable;
  }
```

```
    // Method to set the availability of the book
    public void setAvailable(boolean available) {
        isAvailable = available;
    }
}

// Library class to manage a collection of books
class Library {
    // Array to store Book objects
    private Book[] books;
    private int count;

    // Constructor for the Library class
    public Library(int capacity) {
        books = new Book[capacity];
        count = 0;
    }

    // Method to add a book to the library
    public void addBook(Book book) {
        if (count < books.length) {
            books[count++] = book;
            System.out.println("Book added successfully.");
        } else {
            System.out.println("Library is full.");
        }
    }

    // Method to remove a book from the library by bookID
    public void removeBook(int bookID) {
        for (int i = 0; i < count; i++) {
            if (books[i].getBookID() == bookID) {
                // Shift elements to the left
                for (int j = i; j < count - 1; j++) {
                    books[j] = books[j + 1];
                }
                count--;
```

```
            System.out.println("Book removed successfully.");
            return;
        }
    }
    System.out.println("Book not found.");
  }

  // Method to search for a book by bookID
  public Book searchBook(int bookID) {
    for (int i = 0; i < count; i++) {
      if (books[i].getBookID() == bookID) {
        return books[i];
      }
    }
    return null; // Return null if book not found
  }

  // Method to display all books in the library
  public void displayBooks() {
    if (count == 0) {
      System.out.println("No books available in the library.");
      return;
    }
    for (int i = 0; i < count; i++) {
      System.out.println("Book ID: " + books[i].getBookID());
      System.out.println("Title: " + books[i].getTitle());
      System.out.println("Author: " + books[i].getAuthor());
      System.out.println("Availability: " + (books[i].isAvailable() ? "Available" : "Not
Available"));
      System.out.println();
    }
  }
}

// Main class to demonstrate the library system
public class LibrarySystem {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
```

```
Library library = new Library(5); // Create a library with a capacity of 5 books

while (true) {
   System.out.println("Library System Menu:");
   System.out.println("1. Add Book");
   System.out.println("2. Remove Book");
   System.out.println("3. Search Book");
   System.out.println("4. Display Books");
   System.out.println("5. Exit");
   System.out.print("Choose an option: ");
   int choice = scanner.nextInt();
   scanner.nextLine(); // Consume newline

   switch (choice) {
      case 1: // Add Book
         System.out.print("Enter Book ID: ");
         int bookID = scanner.nextInt();
         scanner.nextLine(); // Consume newline
         System.out.print("Enter Title: ");
         String title = scanner.nextLine();
         System.out.print("Enter Author: ");
         String author = scanner.nextLine();
         System.out.print("Is Available (true/false): ");
         boolean isAvailable = scanner.nextBoolean();
         Book newBook = new Book(bookID, title, author, isAvailable);
         library.addBook(newBook);
         break;

      case 2: // Remove Book
         System.out.print("Enter Book ID to remove: ");
         int removeID = scanner.nextInt();
         library.removeBook(removeID);
         break;

      case 3: // Search Book
         System.out.print("Enter Book ID to search: ");
         int searchID = scanner.nextInt();
         Book foundBook = library.searchBook(searchID);
```

```
            if (foundBook != null) {
                System.out.println("Book found:");
                System.out.println("Book ID: " + foundBook.getBookID());
                System.out.println("Title: " + foundBook.getTitle());
                System.out.println("Author: " + foundBook.getAuthor());
                System.out.println("Availability: " + (foundBook.isAvailable() ? "Available" : "Not
Available"));
            } else {
                System.out.println("Book not found.");
            }
            break;

        case 4: // Display Books
            System.out.println("All Books in the Library:");
            library.displayBooks();
            break;

        case 5: // Exit
            System.out.println("Exiting the Library System.");
            scanner.close();
            return;

        default:
            System.out.println("Invalid option. Please try again.");
      }
    }
  }
}
```

Output:

Library System Menu:
1. Add Book
2. Remove Book
3. Search Book
4. Display Books
5. Exit
Choose an option: 1
Enter Book ID: 101
Enter Title: Physical Science

Enter Author: Einstein
Is Available (true/false): true
Book added successfully.
Library System Menu:
1. Add Book
2. Remove Book
3. Search Book
4. Display Books
5. Exit
Choose an option: 1
Enter Book ID: 102
Enter Title: Biological Science
Enter Author: Subbarao
Is Available (true/false): true
Book added successfully.
Library System Menu:
1. Add Book
2. Remove Book
3. Search Book
4. Display Books
5. Exit
Choose an option: 4
All Books in the Library:
Book ID: 101
Title: Physical Science
Author: Einstein
Availability: Available

Book ID: 102
Title: Biological Science
Author: Subbarao
Availability: Available

Library System Menu:
1. Add Book
2. Remove Book
3. Search Book
4. Display Books

5. Exit
Choose an option: 3
Enter Book ID to search: 102
Book found:
Book ID: 102
Title: Biological Science
Author: Subbarao
Availability: Available
Library System Menu:
1. Add Book
2. Remove Book
3. Search Book
4. Display Books
5. Exit
Choose an option: 2
Enter Book ID to remove: 102
Book removed successfully.
Library System Menu:
1. Add Book
2. Remove Book
3. Search Book
4. Display Books
5. Exit
Choose an option: 4
All Books in the Library:
Book ID: 101
Title: Physical Science
Author: Einstein
Availability: Available

Library System Menu:
1. Add Book
2. Remove Book
3. Search Book
4. Display Books
5. Exit
Choose an option: 5
Exiting the Library System.

Solution-2:

```
package task3;

import java.util.Scanner;

// Interface Taxable with members salesTax and incomeTax and abstract method calcTax
interface Taxable {
   double salesTax = 0.07; // 7%
   double incomeTax = 0.105; // 10.5%

   double calcTax();
}

// Employee class implementing Taxable to calculate incomeTax on yearly salary
class Employee implements Taxable {
   int empId;
   String name;
   double salary;
   // Constructor for Employee class
   public Employee(int empId, String name, double salary) {
      this.empId = empId;
      this.name = name;
      this.salary = salary;
   }

   // Implementation of calcTax() method for Employee class
   @Override
   public double calcTax() {
      return salary * incomeTax;
   }
}

// Product class implementing Taxable to calculate salesTax on unit price of product
class Product implements Taxable {
   int pid;
   double price;
```

```
    int quantity;

    // Constructor for Product class
    public Product(int pid, double price, int quantity) {
        this.pid = pid;
        this.price = price;
        this.quantity = quantity;
    }

    // Implementation of calcTax() method for Product class
    @Override
    public double calcTax() {
        return price * salesTax;
    }
}

// Driver class with main method to accept employee and product information and print
taxes
public class DriverMain {
    public static void main(String[] args) {
        Scanner scanner=new Scanner(System.in);
        // Input employee information from user
        System.out.println("Enter employee details:");
        System.out.print("Employee ID: ");
        int empId = scanner.nextInt();
        System.out.print("Employee Name: ");
        String name = scanner.next();
        System.out.print("Employee Salary: ");
        double salary = scanner.nextDouble();

        // Create Employee object
        Employee employee = new Employee(empId, name, salary);

        // Calculate and print income tax
        System.out.println("Income Tax: " + employee.calcTax());

        // Input product information from user
        System.out. println("Enter product details:");
```

```
        System.out.print("Product ID: ");
        int pid = scanner.nextInt();
        System.out.print("Product Price: ");
        double price = scanner.nextDouble();
        System.out.print("Product Quantity: ");
        int quantity = scanner.nextInt();

        // Create Product object
        Product product = new Product(pid, price, quantity);

        // Calculate and print sales tax
        System.out.println("Sales Tax: " + product.calcTax());
    }
}
```

Output:

```
Enter employee details:
Employee ID: 101
Employee Name: Vamsi
Employee Salary: 100000
Income Tax: 10500.0
Enter product details:
Product ID: 505
Product Price: 5000
Product Quantity: 5
Sales Tax: 350.00000000000006
```