

Skilling Experiment-9

Name: Badisa Naveen

Reg.no: 2000031509

Solid Design Principles with loose coupling and strong cohesion

Case Study: The best book centre company has just purchased a computer for its new automated book inventory management system. You have been asked to program the new system. You are to write an application for the above which contains various classes:

The class descriptions are as follows.

- 1.Book Class: responsible for book details and book search.
 - 2.Inventory View Class: responsible for checking the inventory.
 - 3.Book Discount class: responsible for book discount
 - 4.Book Delivery Class: responsible for online and offline book delivery
 - 5.Book Action Class: responsible book action.
- use Java application but you can apply **SOLID** design principles on it.

Code:

```
package naveen;
public class Book {
    String title;
    String author;
    String getTitle() {
        return title;
    }
    void setTitle(String title) {
        this.title = title;
    }
    String getAuthor() {
        return author;
    }
    void setAuthor(String author) {
        this.author = author;
    }
}
```

```

package naveen;
import java.util.*;
class InventoryView {
    Book book;
    static String searchbook;
    InventoryView(Book book) {
        this.book = book;
    }

    void searchBook(Book b) {
        if(b.getTitle()==searchbook)
            System.out.println("Book found");
        else
            System.out.println("Book is not found");
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Book b = new Book();
        InventoryView i = new InventoryView(b);
        boolean repeat = true;
        while(repeat)
        {
            System.out.println("1.add book\n2.search book\n
others exit");

            switch(sc.nextInt())
            {
                case 1:
                    System.out.println("enter book name and author
name");

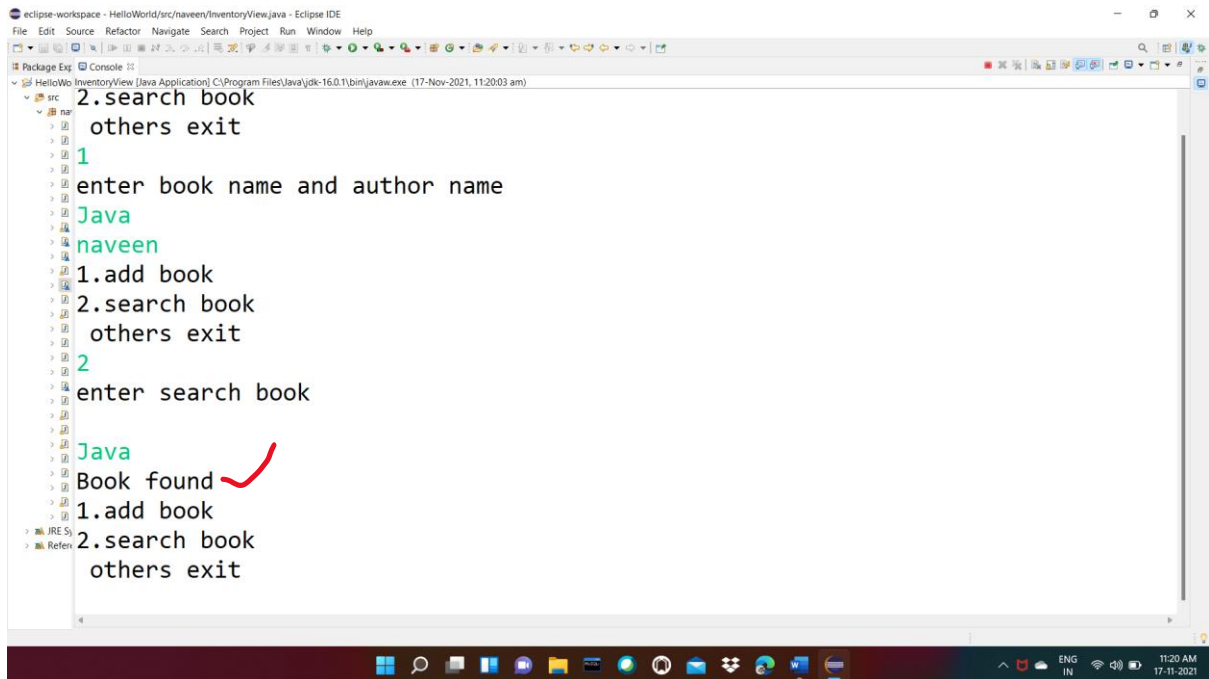
                    b.setTitle(sc.next());
                    b.setAuthor(sc.next());
                    break;
                case 2:
                    System.out.println("enter search book\n");
                    searchbook=sc.next();
                    i.searchBook(b);
                    break;
                default:
                    repeat=false;
            }
        }

        sc.close();}}

```

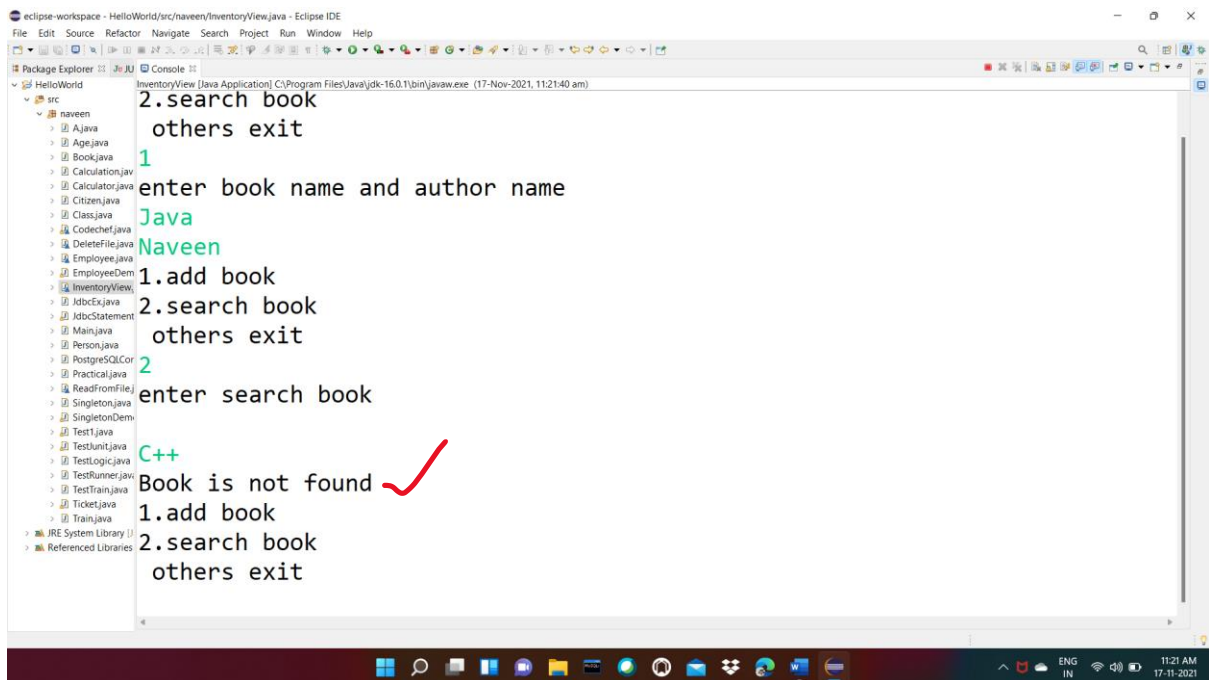
ScreenShots:

Book found:



```
eclipse-workspace - HelloWorld/src/naveen/InventoryView.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Console
HelloWorld [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (17-Nov-2021, 11:20:03 am)
src
  nar
    2.search book
    others exit
    1
    enter book name and author name
    Java
    naveen
    1.add book
    2.search book
    others exit
    2
    enter search book
    Java
    Book found ✓
    1.add book
    2.search book
    others exit
JRE System Libraries
Referenced Libraries
```

Book Not found:



```
eclipse-workspace - HelloWorld/src/naveen/InventoryView.java - Eclipse IDE
File Edit Source Refactor Navigate Search Project Run Window Help
Package Explorer Console
HelloWorld [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe (17-Nov-2021, 11:21:40 am)
src
  naveen
    Aj.java
    Ape.java
    Book.java
    Calculator.java
    Citizen.java
    Class.java
    Codechef.java
    DeleteFile.java
    Employee.java
    EmployeeDem
    InventoryView
    JdbcEx.java
    JdbcStatement
    Main.java
    Person.java
    PostgreSQLCor
    Practical.java
    ReadFromFile
    Singleton.java
    SingletonDem
    Test1.java
    TestLimit.java
    TestLogic.java
    TestRunner.java
    TestTrain.java
    Ticket.java
    Train.java
  JRE System Libraries
  Referenced Libraries
  2.search book
  others exit
  1
  enter book name and author name
  Java
  Naveen
  1.add book
  2.search book
  others exit
  2
  enter search book
  C++
  Book is not found ✓
  1.add book
  2.search book
  others exit
```