



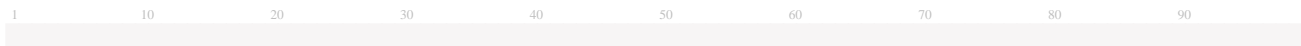
The Report is Generated by DrillBit Plagiarism Detection Software

### Submission Information

Author Name	NNM24IS085
Title	Java task
Paper/Submission ID	4352165
Submitted by	nnm24is076@nmamit.in
Submission Date	2025-09-11 23:25:14
Total Pages, Total Words	7, 359
Document type	Assignment

### Result Information

Similarity **0 %**



### Exclude Information

Quotes	Not Excluded
References/Bibliography	Not Excluded
Source: Excluded < 14 Words	Not Excluded
Excluded Source	<b>0 %</b>
Excluded Phrases	Not Excluded

### Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File





DrillBit Similarity Report

<div>0</div> <div>SIMILARITY %</div>		<div>0</div> <div>MATCHED SOURCES</div>		<div>A</div> <div>GRADE</div>		<div>A-Satisfactory (0-10%)</div> <div>B-Upgrade (11-40%)</div> <div>C-Poor (41-60%)</div> <div>D-Unacceptable (61-100%)</div>	
LOCATION	MATCHED DOMAIN				%	SOURCE TYPE	

```
package library;
```

```
public class Book implements Issueable {
```

```
    private String BookId;
```

```
    private String BookTitle;
```

```
    private String BookAuthor;
```

```
    private boolean Isissued;
```

```
    public Book(String BookId, String BookTitle, String BookAuthor) {
```

```
        this.BookId = BookId;
```

```
        this.BookTitle = BookTitle;
```

```
    this.BookAuthor = BookAuthor;
```

```
        this.Isissued = false;
```

```
    }
```

```
    public String getBookId() {
```

```
        return BookId;
```

```
    }
```

```
    public String BookTitle() {
```

```
        return BookTitle;
```

```
    }
```

```
    public String BookAuthor() {
```

```
        return BookAuthor;
```

```
    }
```

```
    @Override
```

```
public boolean issue() {  
    if (!Isissued) {  
        Isissued = true;  
        return true;  
    }  
    return false;  
}
```

```
@Override  
public boolean returnItem() {  
    if (Isissued) {  
        Isissued = false;  
        return true;  
    }  
    return false;  
}
```

```
@Override  
public boolean isAvailable() {  
    return !Isissued;  
}
```

```
@Override  
public String toString() {  
    return "Book{" +  
        "isbn=" + BookId + "\" +  
        ", title=" + BookTitle + "\" +  
        ", author=" + BookAuthor + "\" +
```

```

        ", issued=" + Isissued +
    '};
}
}

package library;

public interface Issueable {
    boolean issue();
    boolean returnItem();
    boolean isAvailable();
}

package library;

import java.util.ArrayList;
import java.util.List;

public class Systems {
    private List libbooks;

    public Systems() {
        libbooks = new ArrayList<>();
    }

    public void addBook(Book book) {
        libbooks.add(book);
    }

    public Book findBookbyId (String BookId) {

```

```
for (Book book : libbooks) {  
    if (book.getBookId().equals(BookId)) {  
        return book;  
    }  
}  
return null;  
}
```

```
public boolean issueableBook(String BookId) {  
    Book book = findBookbyId(BookId);  
    if (book != null) {  
        return book.issue();  
    }  
    return false;  
}
```

```
public boolean returnBook(String BookId) {  
    Book book = findBookbyId(BookId);  
    if (book != null) {  
        return book.returnItem();  
    }  
    return false;  
}
```

```
public boolean checkAvailable(String BookId) {  
    Book book = findBookbyId(BookId);  
    if (book != null) {  
        return book.isAvailable();  
    }  
}
```

```
    }  
    return false;  
}  
  
public void displayAllBooks() {  
    for (Book book : libbooks) {  
        System.out.println(book);  
    }  
}  
}  
  
package library;  
  
import org.junit.jupiter.api.BeforeEach;  
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.*;  
  
public class LibraryTest {  
    private Book book;  
  
    @BeforeEach  
    public void setUp() {  
        book = new Book("201", "Data Structures", "Robert Lafore");  
    }  
  
    @Test  
    public void testInitialAvailability() {  
        assertTrue(book.isAvailable(), "Book should be available when first created");  
    }  
}
```

```
}
```

```
@Test
```

```
public void testIssueBook() {  
    assertTrue(book.issue(), "First issue should succeed");  
    assertFalse(book.issue(), "Second issue should fail since already issued");  
}
```

```
@Test
```

```
public void testReturnBook() {  
    book.issue(); // first issue  
    assertTrue(book.returnItem(), "Return should succeed after issue");  
    assertFalse(book.returnItem(), "Second return should fail since already returned");  
}
```

```
@Test
```

```
public void testBookDetails() {  
    assertEquals("201", book.getBookId());  
    assertEquals("Data Structures", book.BookTitle());  
    assertEquals("Robert Lafore", book.BookAuthor());  
}
```

```
@Test
```

```
public void testToStringFormat() {  
    String output = book.toString();  
    assertTrue(output.contains("201"));  
    assertTrue(output.contains("Data Structures"));  
    assertTrue(output.contains("Robert Lafore"));  
}
```



}

}