

## Main.java



Share

Run

```
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3 package models;
4
5 import java.io.Serializable;
6 import java.time.LocalDate;
7
8 /**
9  * Represents a borrowing transaction between a Patron and a Book.
10 * This class links the three main entities (Book, Patron, and the
11 * time data).
12 public class Transaction implements Serializable {
13     private static final long serialVersionUID = 1L;
14 }
```

## Main.java

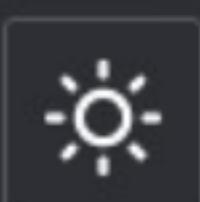


Share

Run

```
14  
15     private String transactionId;  
16     private String bookIsbn;  
17     private String patronId;  
18     private LocalDate borrowDate;  
19     private LocalDate dueDate;  
20     private LocalDate returnDate; // Null if not yet returned  
21  
22     // Constructor for a new checkout  
23     public Transaction(String transactionId, String bookIsbn,  
24         String patronId, LocalDate borrowDate, LocalDate dueDate) {  
25         this.transactionId = transactionId;  
26         this.bookIsbn = bookIsbn;  
27         this.patronId = patronId;  
28         this.borrowDate = borrowDate;
```

## Main.java

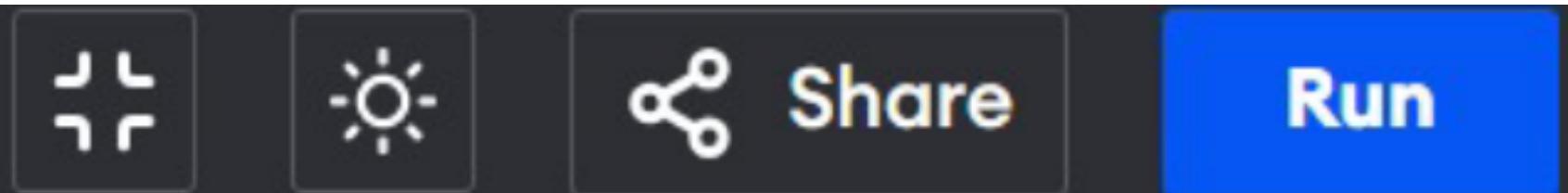


Share

Run

```
27     this.borrowDate = borrowDate;
28     this.dueDate = dueDate;
29     this.returnDate = null; // Initially not returned
30 }
31
32 // Getters
33 public String getTransactionId() { return transactionId; }
34 public String getBookIsbn() { return bookIsbn; }
35 public String getPatronId() { return patronId; }
36 public LocalDate getBorrowDate() { return borrowDate; }
37 public LocalDate getDueDate() { return dueDate; }
38 public LocalDate getReturnDate() { return returnDate; }
39 public boolean isReturned() { return returnDate != null; }
40
41 public boolean isOverdue() {
42     return !isReturned() && LocalDate.now().isAfter(dueDate);
```

## Main.java



```
41     return !isReturned() && LocalDate.now().isAfter(dueDate);
42 }
43
44 // Setter for return
45 public void setReturnDate(LocalDate returnDate) {
46     this.returnDate = returnDate;
47 }
48
49 @Override
50 public String toString() {
51     String status = isReturned() ? "Returned" : (isOverdue() ?
52         "OVERDUE" : "Active");
53     return String.format("TXN ID: %s | ISBN: %s | Patron ID: %s
| Due: %s | Status: %s",
54         transactionId, bookIsbn, patronId, dueDate.toString()
```

## Main.java



Share

Run

```
44     // Setter for return
45     public void setReturnDate(LocalDate returnDate) {
46         this.returnDate = returnDate;
47     }
48
49     @Override
50     public String toString() {
51         String status = isReturned() ? "Returned" : (isOverdue() ?
52             "OVERDUE" : "Active");
53         return String.format("TXN ID: %s | ISBN: %s | Patron ID: %s
54             | Due: %s | Status: %s",
55             transactionId, bookIsbn, patronId, dueDate.toString(),
56             (), status);
57     }
58 }
```