**CSCI 391 – ST: Microservices, Lab 1 – Project Lombok**

Original Book Class

Here’s the original Book class with a few common Java patterns:

public class Book {

private String title;

private String author;

private double price;

public Book(String title, String author, double price) {

this.title = title;

this.author = author;

this.price = price;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

@Override

public String toString() {

return "Book{" +

"title='" + title + '\'' +

", author='" + author + '\'' +

", price=" + price +

'}';

}

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (o == null || getClass() != o.getClass()) return false;

Book book = (Book) o;

return Double.compare(book.price, price) == 0 &&

title.equals(book.title) &&

author.equals(book.author);

}

@Override

public int hashCode() {

return Objects.hash(title, author, price);

}

}

### Applying Lombok Annotations

Now, it's time to simplify the Book class by using [Lombok annotations](https://medium.com/javarevisited/all-the-16-lombok-annotations-explained-in-a-4-minute-article-926f71934ec6). Please first read through the attached article to understand what Lombok annotations are, and how we use them.

### Refactor Book Class Using Lombok

Here’s how the class would look after applying Lombok annotations:

TODO!! Refactor the book class to use [Lombok annotations](https://medium.com/javarevisited/all-the-16-lombok-annotations-explained-in-a-4-minute-article-926f71934ec6). Try to use the least number of lines of code as possible ***without using the @Data annotation***. Your submission should be a single java file Book.java and should be uploaded to Canvas.

### Example Usage

Here’s how you can use the Book class with the generated methods:

public class Main {

public static void main(String[] args) {

// Using the Builder pattern to create a Book object

Book book1 = Book.builder()

.title("Effective Java")

.author("Joshua Bloch")

.price(45.99)

.build();

// Using the generated toString() method

System.out.println(book1.toString());

// Creating another Book instance

Book book2 = new Book("Clean Code", "Robert C. Martin", 35.50);

// Using the generated equals() and hashCode() methods

System.out.println(book1.equals(book2)); // false

}

}

Conclusion

In this exercise, you have:

* Simplified the Book class by using Lombok annotations.
* Reduced boilerplate code, making your class more concise and readable.
* Learned how to use several powerful [Lombok annotations](https://medium.com/javarevisited/all-the-16-lombok-annotations-explained-in-a-4-minute-article-926f71934ec6).

Now, practice applying these annotations to other classes in your project (will do this in upcoming labs), and enjoy cleaner, more efficient code!