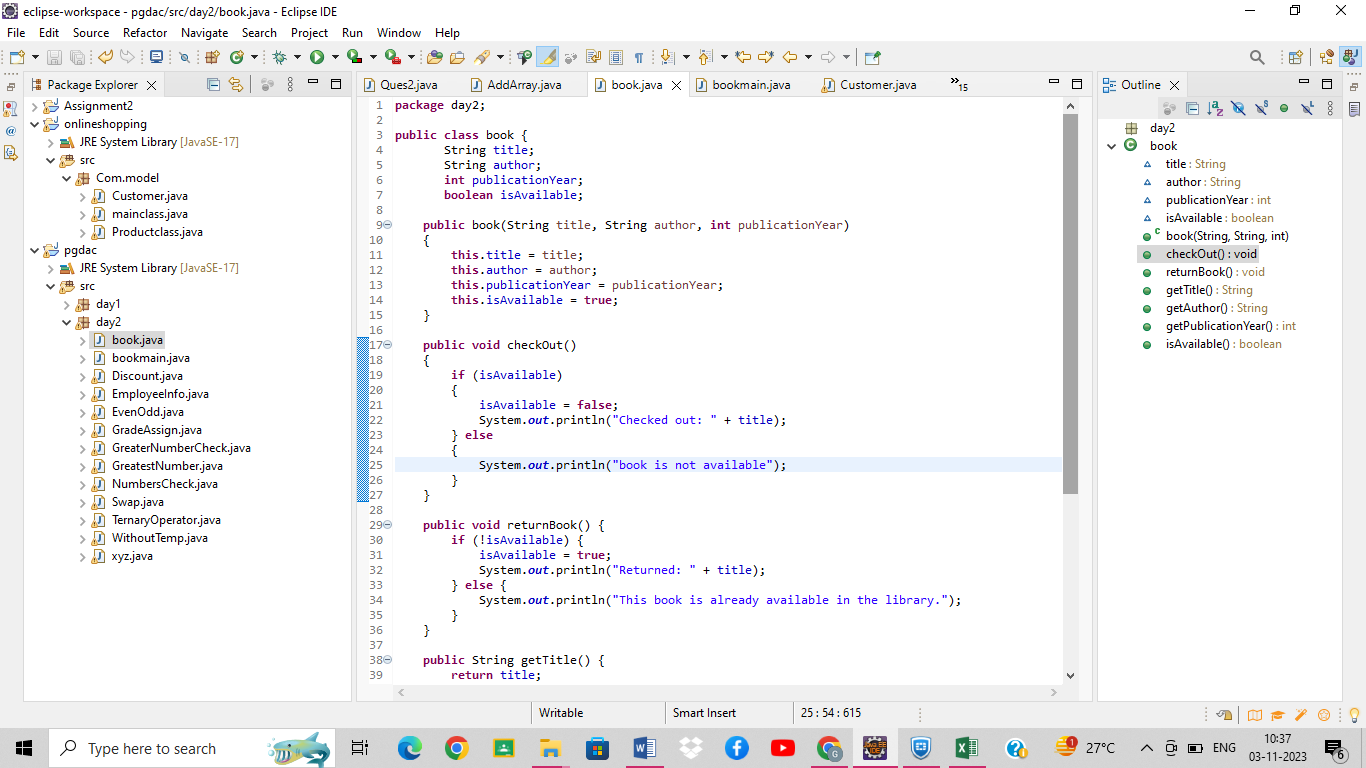
Imagine you are developing a simple library management system. Create a Java class called Book that represents a book in the library. The Book class should have the following attributes:

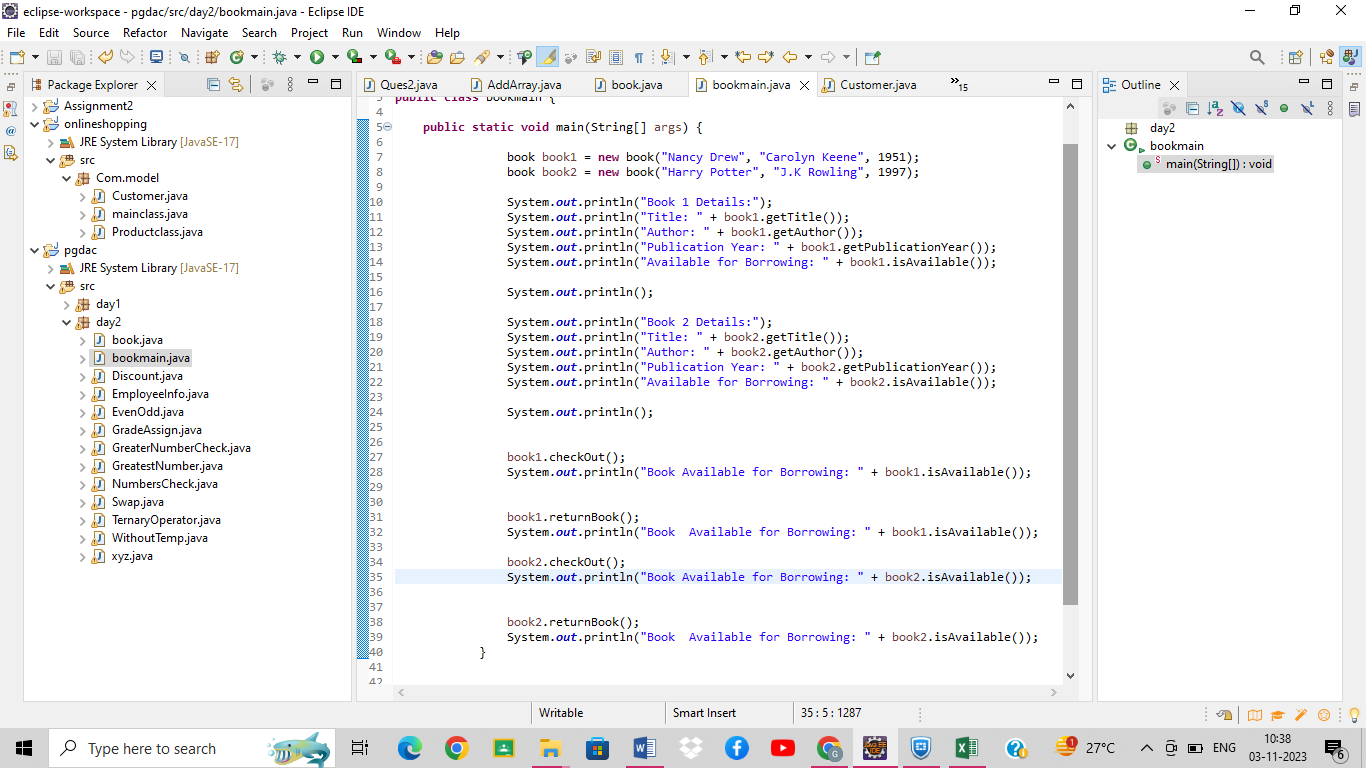
* title (String) - the title of the book.
* author (String) - the author of the book.
* publicationYear (int) - the year the book was published.
* isAvailable (boolean) - a flag to check if the book is available for borrowing.

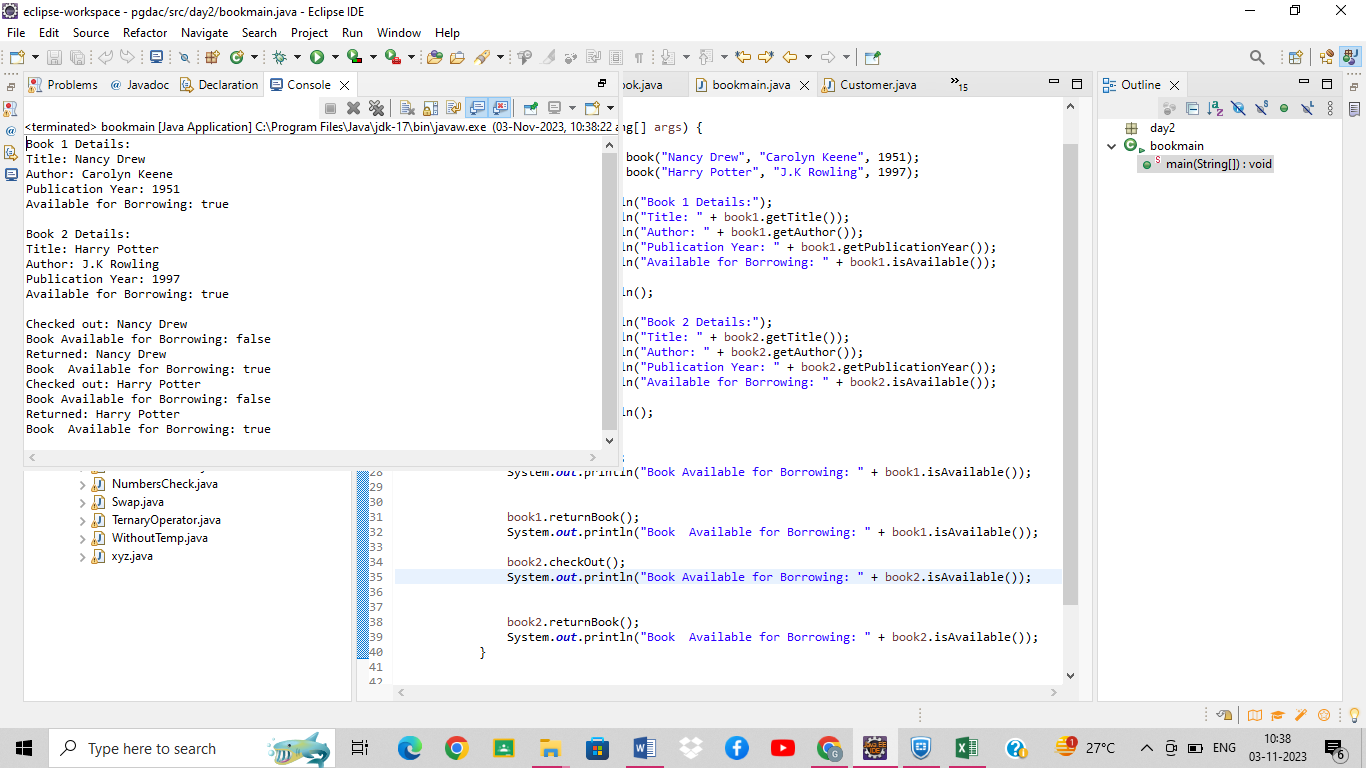
The Book class should also have the following methods:

* checkOut() - a method to check out the book, which sets the isAvailable attribute to false.
* returnBook() - a method to return the book, which sets the isAvailable attribute to true.

Next, create two instances of the Book class in your Main class. Set their attributes, check one out, and then return it. Finally, print the details of both books, including whether they are available for borrowing.







 You are tasked with creating a class hierarchy to represent different types of vehicles. The base class is Vehicle, and you should create two subclasses, Car and Bicycle. Each class should have attributes and methods specific to the type of vehicle.

Vehicle Class:

Attributes:

make (String) - the make or manufacturer of the vehicle.

model (String) - the model of the vehicle.

year (int) - the year of manufacture.

Methods:

A constructor to initialize the attributes.

start() - a method that prints "The vehicle is starting."

Car Class:

Additional Attributes:

numDoors (int) - the number of doors in the car.

Additional Methods:

A constructor to initialize the make, model, year, and numDoors.

drive() - a method that prints "The car is moving forward."

Bicycle Class:

Additional Attributes:

numGears (int) - the number of gears on the bicycle.

Additional Methods:

A constructor to initialize the make, model, year, and numGears.

pedal() - a method that prints "The bicycle is in motion."

Create instances of the Car and Bicycle classes in your Main class. Set their attributes and call the start(), drive(), and pedal() methods to demonstrate inheritance and method overriding.

