Vehicle Rental System

You are designing a Vehicle Rental System for a rental company that manages different types of vehicles (e.g., cars and bikes). Each vehicle has some common attributes, but their behaviors differ depending on the type of vehicle.

Implement the following requirements:
1. Base Class
Create a base class Vehicle with:
Private attributes: registrationNumber (string) and brand (string).
Protected attributes: rentalRate (float).
A public method displayDetails() to show common details of the vehicle.

2. Inheritance
Derive two classes, Car and Bike, from the Vehicle class.
3. Overriding
Override the displayDetails() method in both derived classes to include specific attributes:

For Car: Add numberOfDoors (integer).
For Bike: Add isElectric (boolean).

4. Overloading
Implement a method calculateRentalCost() in both derived classes using function overloading:
For Car: Overload it to accept days (integer) and return the rental cost as
days * rentalRate.

For Bike: Overload it to accept hours (integer) and return the rental cost as
hours * (rentalRate / 24).

5. Access Modifier Usage
Ensure registrationNumber and brand are only accessible within the base class.
Allow rentalRate to be accessible in the derived classes.

6. Implementation
Create instances of Car and Bike in the main function, set their attributes, display their details, and calculate their rental costs.

Imagine you are designing a library management system, where you have the following classes:

- `Item`: A general class representing items in the library.
- `Book` and `Magazine`: Both are specific types of items.
- `EBook` and `OnlineMagazine`: Both inherit from `Book` and `Magazine`, respectively.

There will be arise an ambiguity. How can you possibly solve that?