

(3 marks)

Write a class named Medicine and a class named PrescriptionMedicine extended from Medicine (i.e. Medicine is a superclass and PrescriptionMedicine is a subclass) with the following information:

Medicine
- name : String - indication:String - expirationYear : int
+ Medicine() + Medicine(name:String, indication:String, expirationYear:int) + getName():String + getIndication (): String + getExpirationYear(): int + toString(): String

Where:

- Medicine() - default constructor
- Medicine(name:String, indication:String, expirationYear:int) – parameterized constructor, which sets values to name, indication, and expirationYear
- getName (): String – return medicine name
- getIndication (): String – return indication with the first character in uppercase, the remaining characters are lowercase
- getExpirationYear(): int – return the expiration year
- toString():String - return a string format that contains all the information of the Medicine: name, indication, expirationYear

PrescriptionMedicine
- doctorName : String
+ PrescriptionMedicine() + PrescriptionMedicine(name:String, indication:String, expirationYear:int, doctorName:String) + getters & setters + isExpired(currentYear:int): String + toString(): String

Where:

- PrescriptionMedicine () - default constructor
- PrescriptionMedicine(name:String, indication:String, expirationYear:int, doctorName:String) - parameterized constructor, which sets values to name, indication, expirationYear, and doctorName
- getters & setters: write the setters and getters all of the fields
- isExpired(currentYear:int): String – return "Expired" if the current year is greater than the expiration year, otherwise return "Valid".
- toString(): String - Override toString() method to return a string that contains all the information of the PrescriptionMedicine: name, indication, expirationYear, doctorName. The name, indication and doctorName with the first character in uppercase, the remaining characters are lowercase

The program output might look something like this

Enter the medicine: panadol Enter the indication: headache Enter the expiration year: 2025 Enter the doctor's name: david 1. Test getIndication() 2. Test isExpired() 3. Test toString() Enter TC(1 2 3): 1 OUTPUT: Headache	Enter the name: panadol Enter the indication: headache Enter the expiration year: 2025 Enter the doctor's name: david 1. Test getIndication() 2. Test isExpired() 3. Test toString() Enter TC(1 2 3): 2 Enter current year: 2024 OUTPUT: Valid	Enter the name: panadol Enter the indication: headache Enter the expiration year: 2025 Enter the doctor's name: davID 1. Test getIndication() 2. Test isExpired() 3. Test toString() Enter TC(1 2 3): 3 OUTPUT: Panadol, Headache, 2025, David
---	--	---

