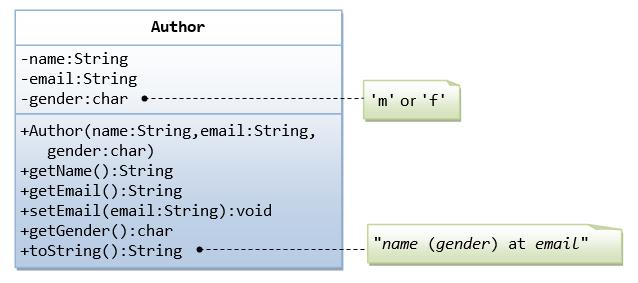
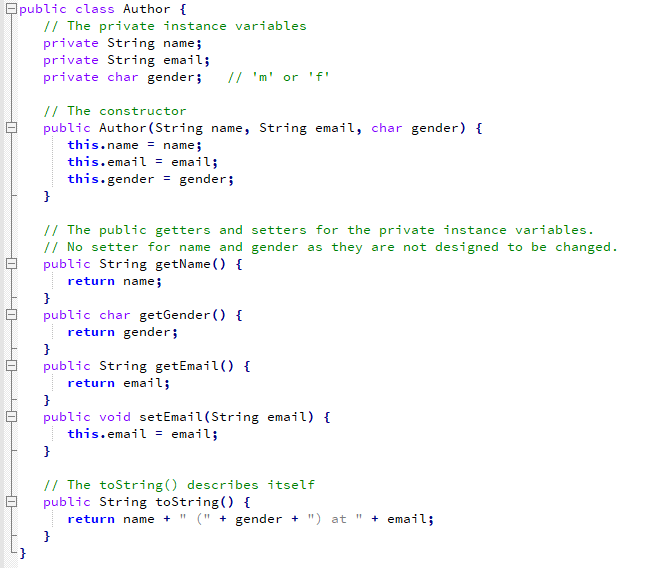
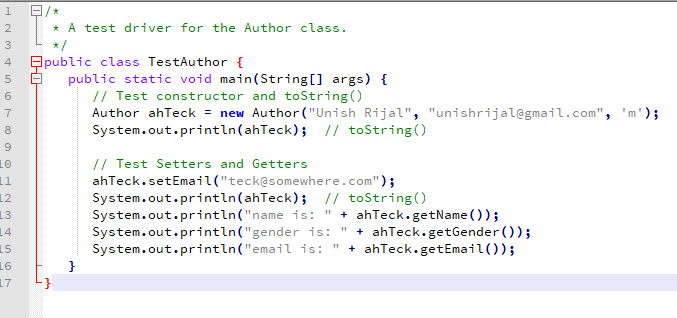
Q1. Convert this UML class diagram to java code and test the java code.

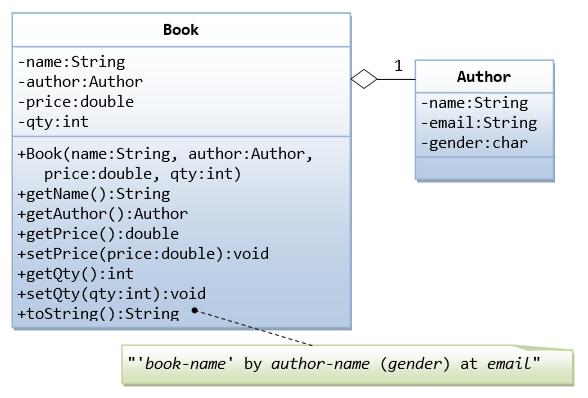


**The Author Class (Author.java)**

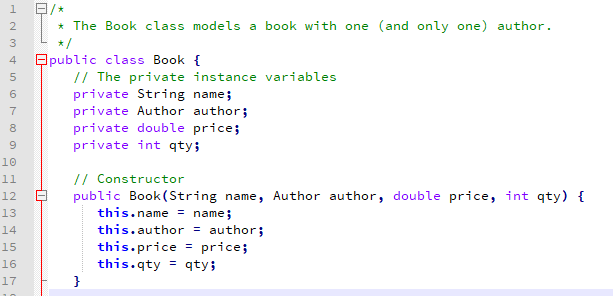


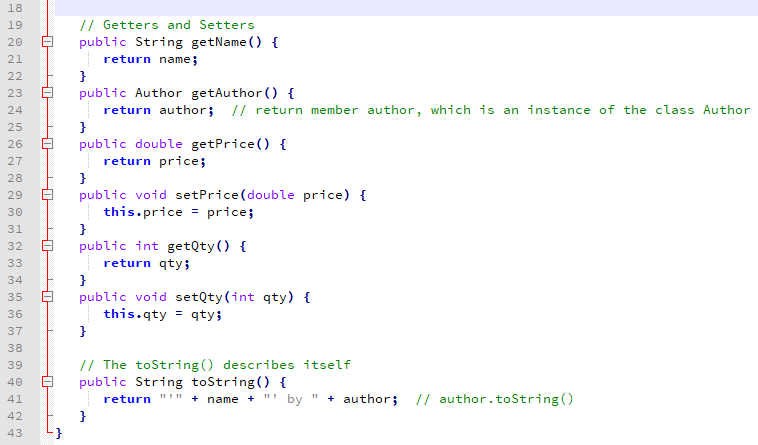
**A Test Driver for The Author Class (TestAuthor.java)**



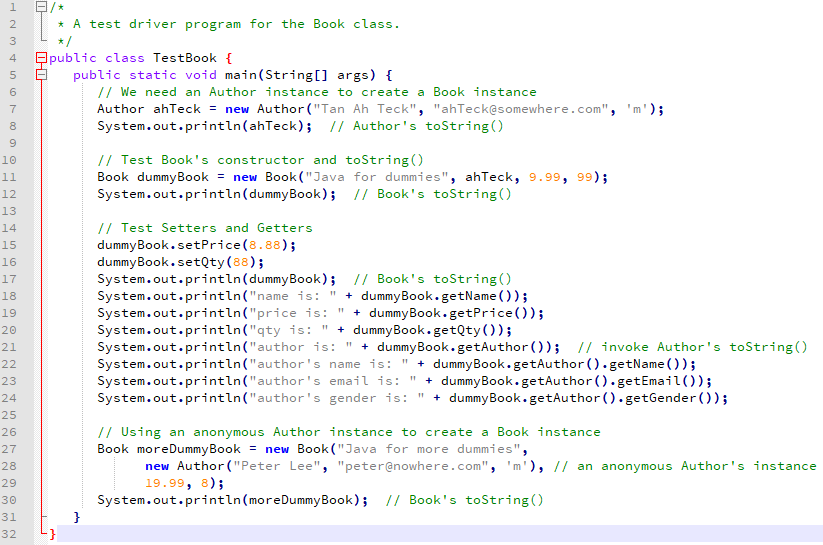
Q2. Convert this UML class diagram to java code and test the java code.****

**The Book Class (Book.java)**

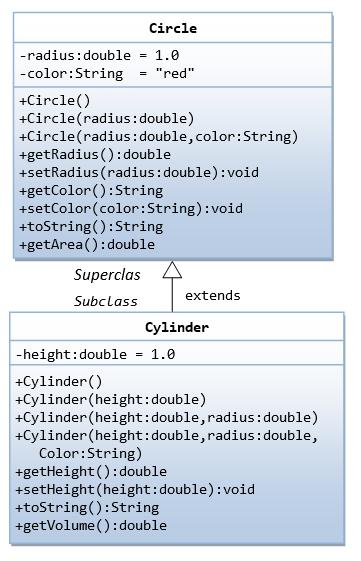




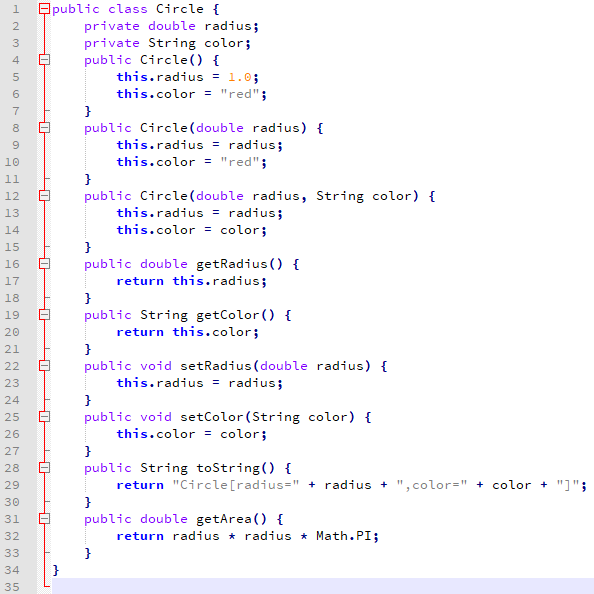
**A Test Driver For Program For The Book Class (TestBook.java)**



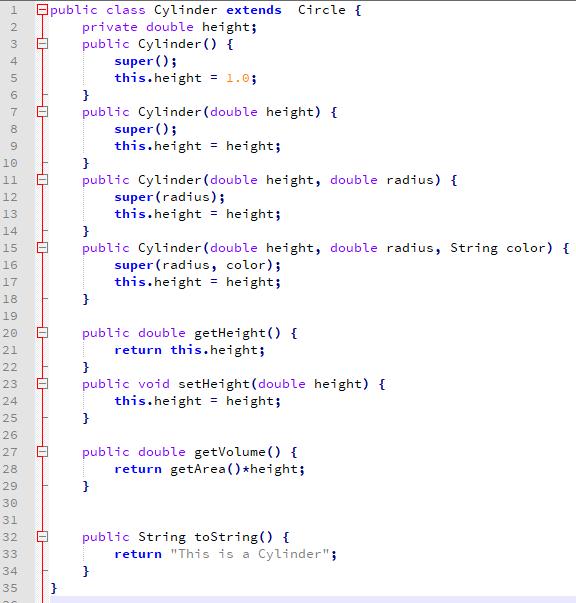
Q3. Using **inheritance** convert the following UML class diagram to java code and test the java code.



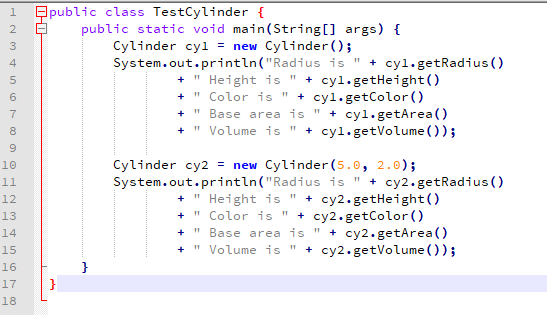
**The Circle Class (Circle.java)**



**The Cylinder Class (Cylinder.java)**



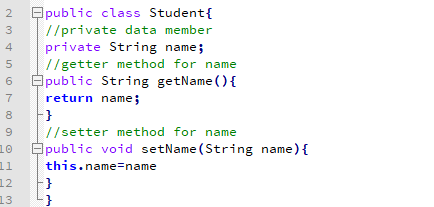
**A Test Driver For Program For The CylinderTest Class (CylinderTest.java)**



4. Encapsulation is defined as keeping the data and method in a class so that it cannot be accessed by the user directly but can be called using public methods. It helps for data hiding in java because other class cannot access data through private data members. In the example below class student is a encapsulated class that has only one field with its getter and setter. Then the value is passed in class student by creating instance of the encapsulated class in main class Test.

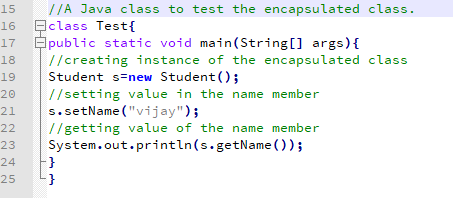
**Example:**

**The Student Class (Student.java)**



The public setName() and getName() methods are the access points of the instance variables of the Student class. Normally, these methods are referred as getters and setters. Therefore, any class that wants to access the variables should access them through these getters and setters.

The variable of the Student Class can be accessed Using the Following Program:-

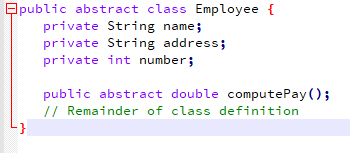


5. Abstraction is defined as the method of hiding the implementation details and showing only functionality the user. Abstraction focuses on what the object does rather how it does it.

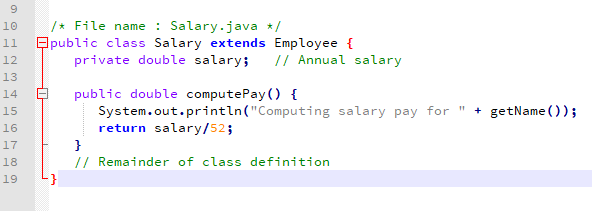
The method which is declared as abstract and does not have implementation is known as an abstract method.

In this example, Employee is an abstract class that contains one abstract method computePay() and its implementation is provided by Salary class.

**The Abstract Class Employee**



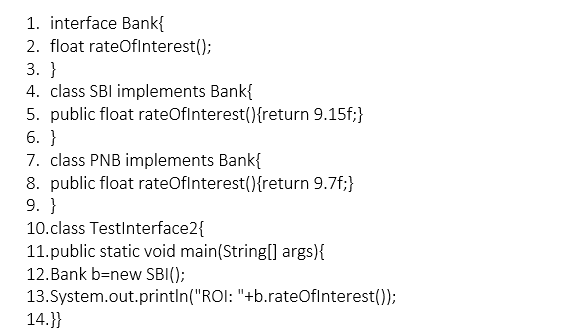
Suppose Salary Class inherits the Employee class, then it should implement the ComputePay() method as shown as below;



6. An interface in java is defined as a blueprint of a class. It can contain abstract methods and variables but it cannot have method body. It provides loose coupling.

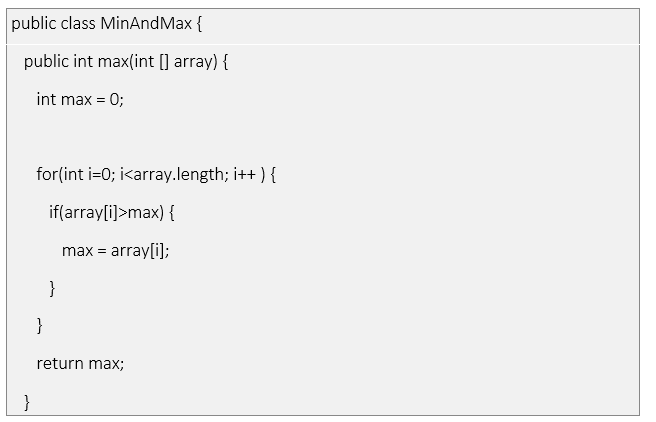
**Example:**

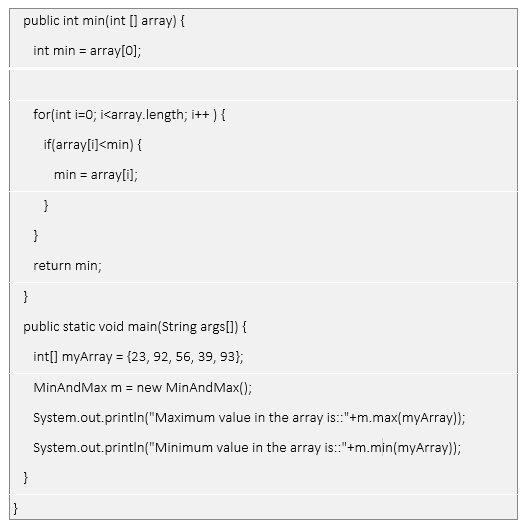
**The Interface Animal**



****

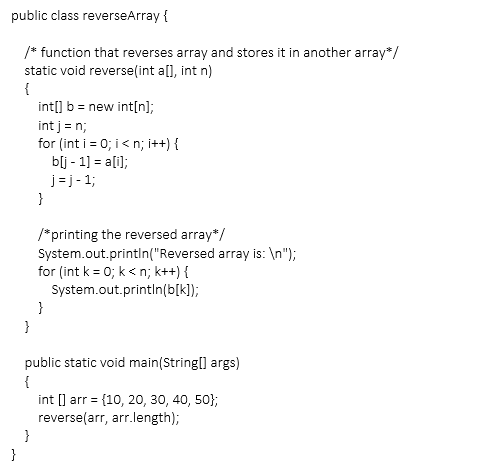
# 7.

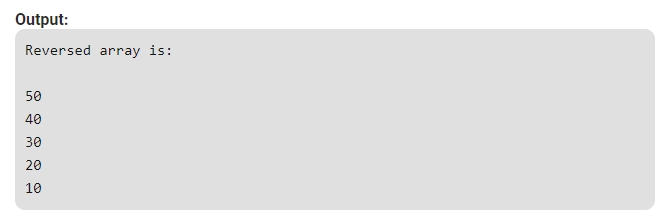




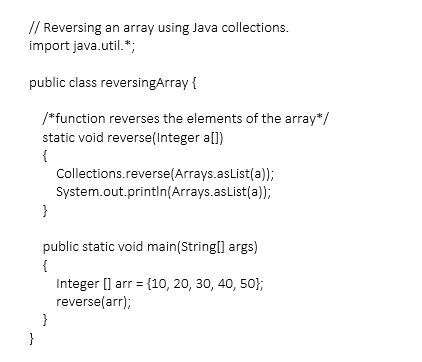


# Q8.



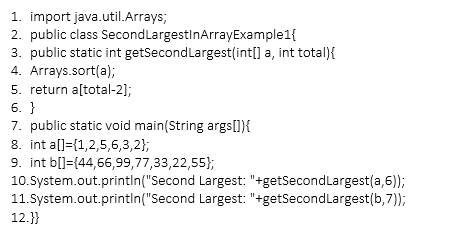


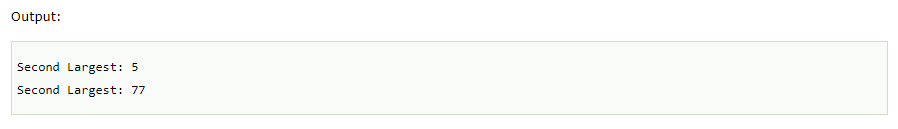
**Method 2**

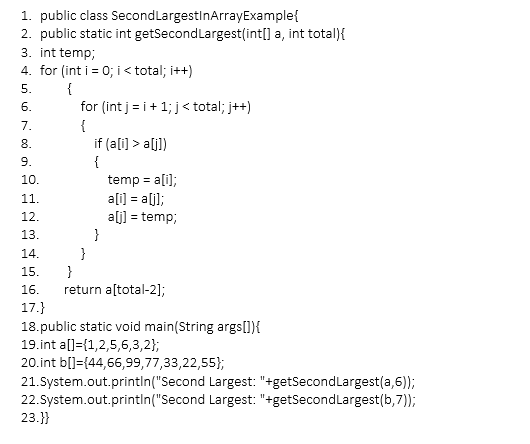


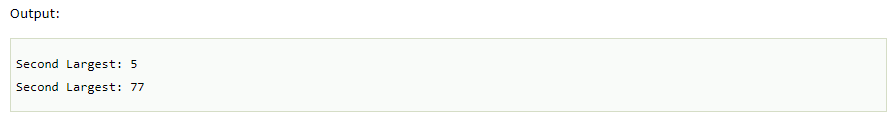


# Q9.









10. The Array List class extends Abstract List and implements the List interface. It supports dynamic arrays that can grow as needed.

