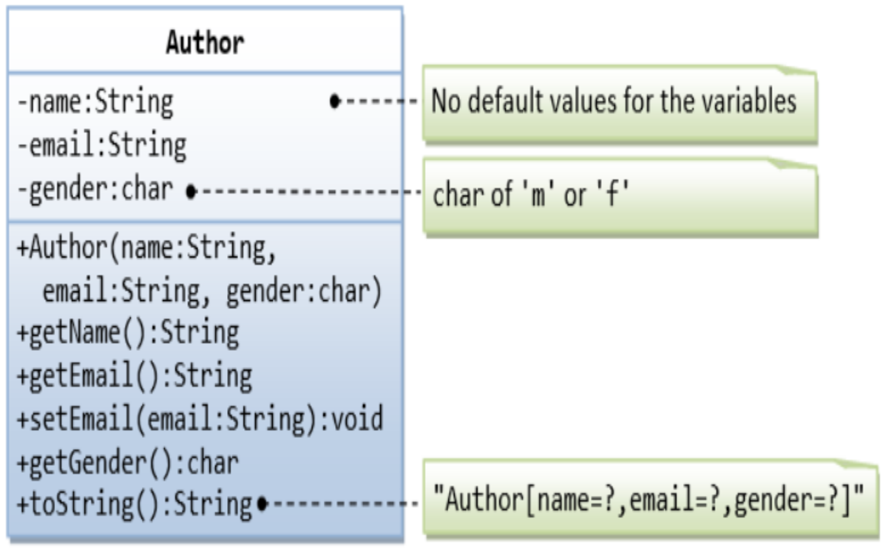
This first exercise shall lead you through all the concepts involved in OOP Composition.


Author
-name:String
-email:String
-gender:char
+Author(name:String, email:String, gender:char)
+getName():String
+getEmail():String
+setEmail(email:String):void
+getGender():char
+toString():String
No default values for the variables
char of 'm' or 'f'
"Author[name=?,email=?,gender=?]"

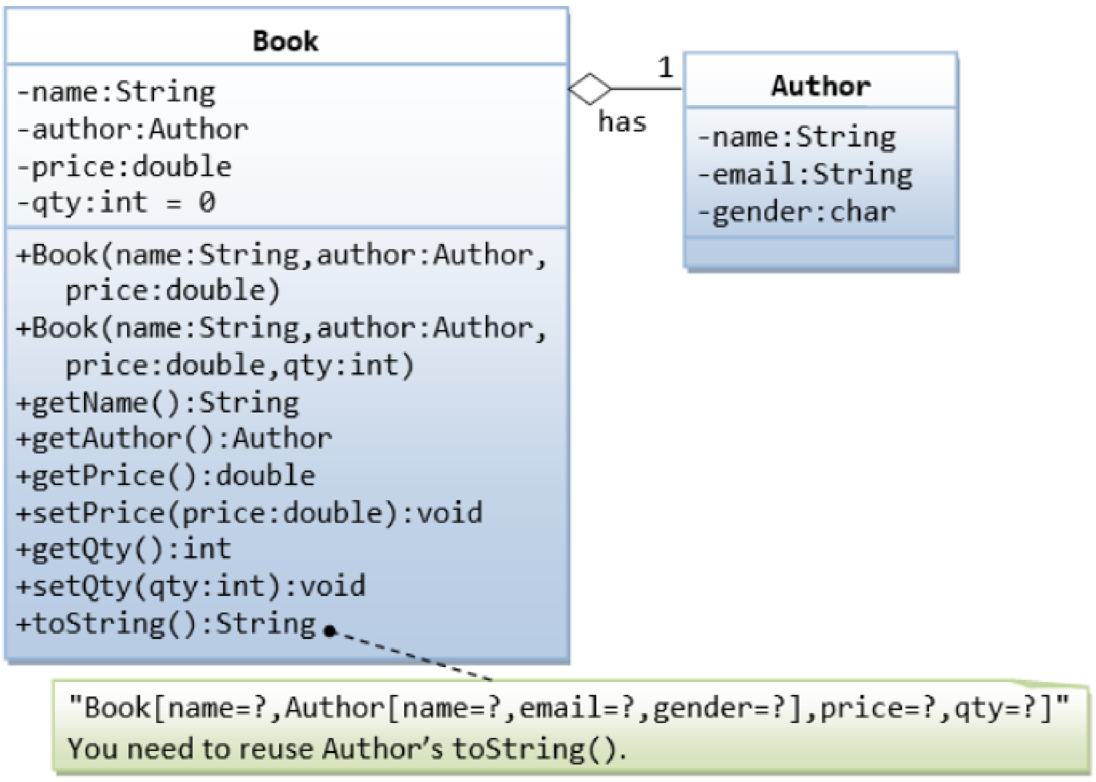
A class called Author (as shown in the class diagram) is designed to model a book's author. It contains:

- Three private instance variables: name (String), email (String), and gender (char of either 'm' or 'f');
- One constructor to initialize the name, email and gender with the given values;

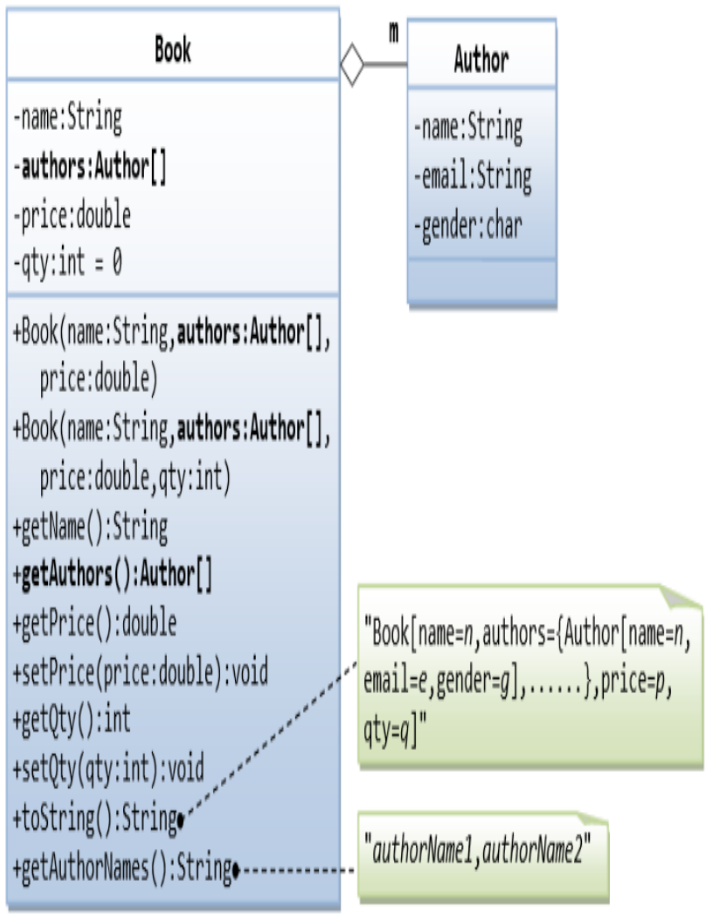
  ```
  public Author (String name, String email, char gender) {......}
  ```

  (There is no default constructor for Author, as there are no defaults for name, email and gender.)
- public getters/setters: getName(), getEmail(), setEmail(), and getGender();
  (There are no setters for name and gender, as these attributes cannot be changed.)
- A toString() method that returns "Author[name=?,email=?,gender=?]", e.g., "Author[name=Tan Ah Teck,email=ahTeck@somewhere.com,gender=m]".

Write the Author class. Also write a *test driver* called TestAuthor to test all the public methods, e.g.,



"Book[name=?,Author[name=?,email=?,gender=?],price=?,qty=?]"
You need to reuse Author's toString().


Book
-name:String
-authors:Author[]
-price:double
-qty:int = 0
+Book(name:String,authors:Author[], price:double)
+Book(name:String,authors:Author[], price:double,qty:int)
+getName():String
+getAuthors():Author[]
+getPrice():double
+setPrice(price:double):void
+getQty():int
+setQty(qty:int):void
+toString():String
+getAuthorNames():String
m
Author
-name:String
-email:String
-gender:char
"Book[name=*n*,authors={Author[name=*n*, email=*e*,gender=*g*],......},price=*p*, qty=*q*]"
"*authorName1*,*authorName2*"

In the earlier exercise, a book is written by one and only one author. In reality, a book can be written by one or more author. Modify the Book class to support one or more authors by changing the instance variable `authors` to an `Author` array.

Notes:

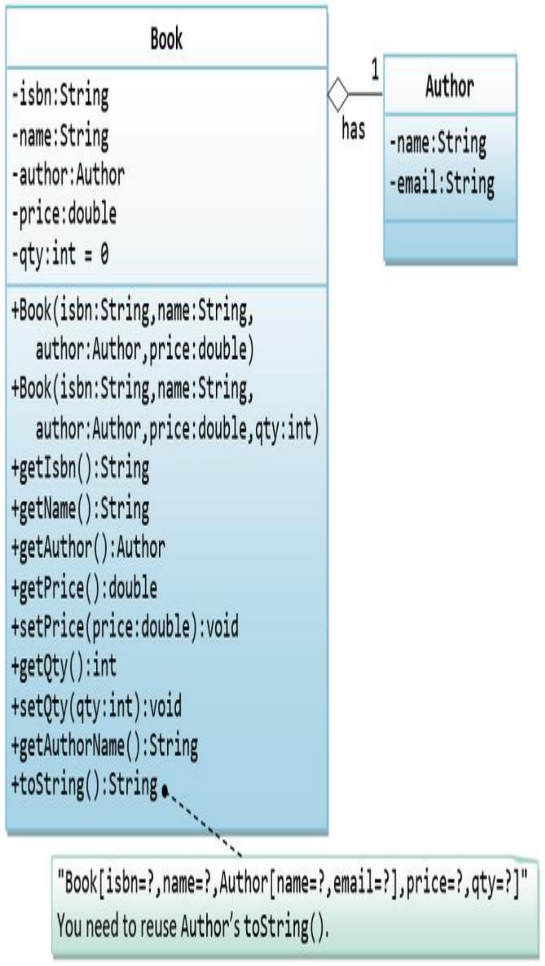
- The constructors take an array of `Author` (i.e., `Author[]`), instead of an `Author` instance. In this design, once a Book instance is constructor, you cannot add or remove author.
- The `toString()` method shall return "`Book[name=?,authors={Author[name=?,email=?,gender=?],......},price=?,qty=?]`".

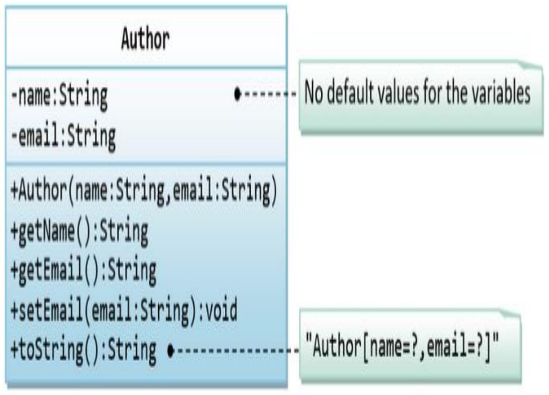
You are required to:

1. Write the code for the Book class. You shall re-use the `Author` class written earlier.
2. Write a test driver (called `TestBook`) to test the Book class.

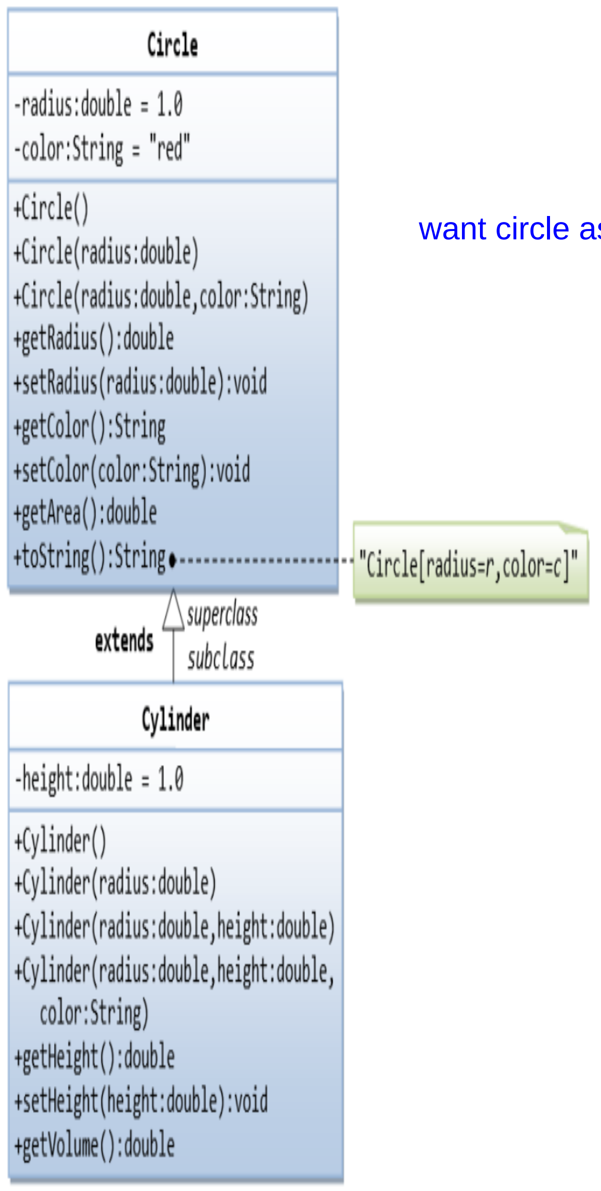
A class called Author, which models an author of a book, is designed as shown in the class diagram. A class called Book, which models a book written by ONE author and composes an instance of Author as its instance variable, is also shown. Write the Author and Book classes.

now in session


Book
-isbn:String
-name:String
-author:Author
-price:double
-qty:int = 0
+Book(isbn:String,name:String, author:Author,price:double)
+Book(isbn:String,name:String, author:Author,price:double,qty:int)
+getIsbn():String
+getName():String
+getAuthor():Author
+getPrice():double
+setPrice(price:double):void
+getQty():int
+setQty(qty:int):void
+getAuthorName():String
+toString():String
has
1
Author
-name:String
-email:String
"Book[isbn=?,name=?,Author[name=?,email=?],price=?,qty=?]"
You need to reuse Author's toString().



This exercise shall guide you through the important concepts in inheritance.



want circle as trait

In this exercise, a subclass called `Cylinder` is derived from the superclass `Circle` as shown in the class diagram (where an an arrow pointing up from the subclass to its superclass). Study how the subclass `Cylinder` invokes the superclass' constructors (via `super()` and `super(radius)`) and inherits the variables and methods from the superclass `Circle`.

You can reuse the `Circle` class that you have created in the previous exercise. Make sure that you keep "`Circle.class`" in the same directory.