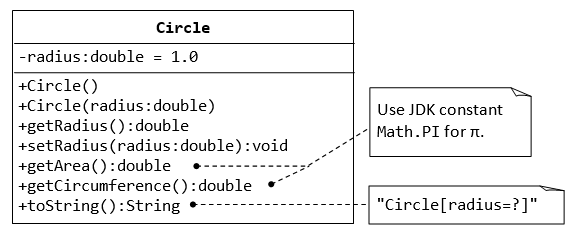
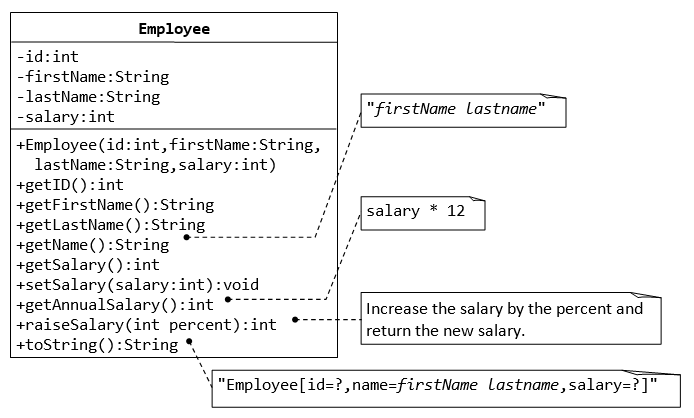
#### 1.2  1.2  Ex: A Simplified Circle Class

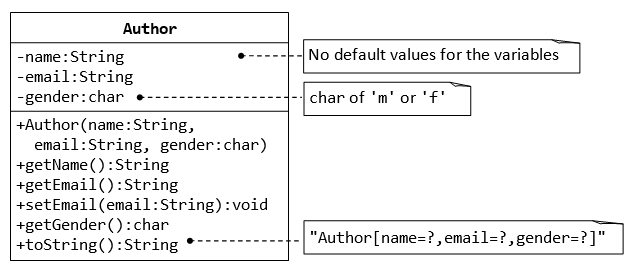


#### 1.4  1.4  Ex: The Employee Class



**2.1  2.1  Ex: The Author and Book Classes (An Introduction to OOP Composition)**

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



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

Author ahTeck = new Author("Tan Ah Teck", "ahteck@nowhere.com", 'm'); // Test the constructor

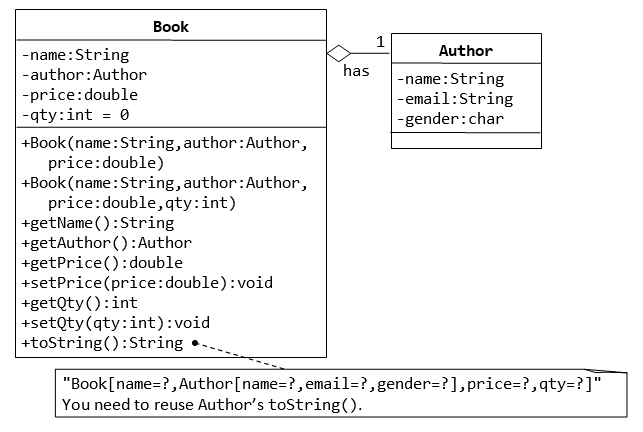
System.out.println(ahTeck); // Test toString()

ahTeck.setEmail("paulTan@nowhere.com"); // Test setter

System.out.println("name is: " + ahTeck.getName()); // Test getter

System.out.println("email is: " + ahTeck.getEmail()); // Test getter

System.out.println("gender is: " + ahTeck.getGender());

b)   


Write the Book class (which uses the Author class written earlier). Also write a test driver called TestBook to test all the public methods in the class Book. Take Note that you have to construct an instance of Author before you can construct an instance of Book. E.g.,

// Construct an author instance

Author ahTeck = new Author("Tan Ah Teck", "ahteck@nowhere.com", 'm');

System.out.println(ahTeck); // Author's toString()

Book dummyBook = new Book("Java for dummy", ahTeck, 19.95, 99); // Test Book's Constructor

System.out.println(dummyBook); // Test Book's toString()

// Test Getters and Setters

dummyBook.setPrice(29.95);

dummyBook.setQty(28);

System.out.println("name is: " + dummyBook.getName());

System.out.println("price is: " + dummyBook.getPrice());

System.out.println("qty is: " + dummyBook.getQty());

System.out.println("Author is: " + dummyBook.getAuthor()); // Author's toString()

System.out.println("Author's name is: " + dummyBook.getAuthor().getName());

System.out.println("Author's email is: " + dummyBook.getAuthor().getEmail());

// Use an anonymous instance of Author to construct a Book instance

Book anotherBook = new Book("more Java",

new Author("Paul Tan", "paul@somewhere.com", 'm'), 29.95);

System.out.println(anotherBook); // toString()

TRY:

1. Printing the name and email of the author from a Book instance. (Hint: aBook.getAuthor().getName(), aBook.getAuthor().getEmail()).
2. Introduce new methods called getAuthorName(), getAuthorEmail(), getAuthorGender() in the Book class to return the name, email and gender of the author of the book. For example,

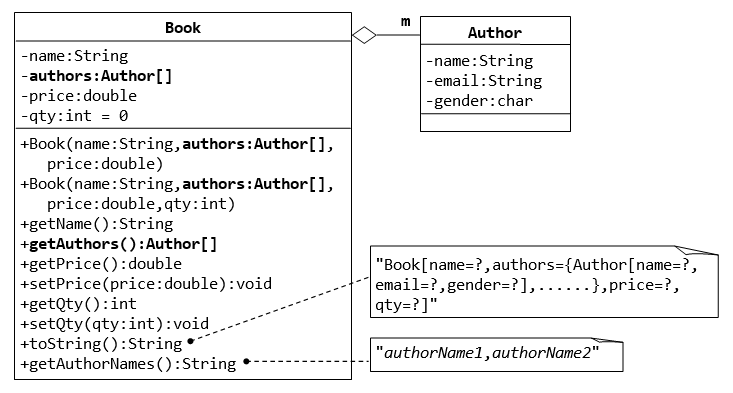
public String getAuthorName() {

return this.author.getName();

// cannot use author.name as name is private in Author class

}

**2.2  2.2  Exercise (Advanced): Book and Author Classes Again - An Array of Objects as an Instance Variable**



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.

**Hints**:

// Declare and allocate an array of Authors

Author[] authors = new Author[2];

authors[0] = new Author("Tan Ah Teck", "AhTeck@somewhere.com", 'm');

authors[1] = new Author("Paul Tan", "Paul@nowhere.com", 'm');

// Declare and allocate a Book instance

Book javaDummy = new Book("Java for Dummy", authors, 19.99, 99);

System.out.println(javaDummy); // toString()