

Siksha 'O' Anusandhan Deemed to be University
Institute of Technical Education and Research
Data Structure and Algorithms (CSE-2001)
Minor Project Feb'2020

Programme: B.Tech(All Branches)
Full marks: 15

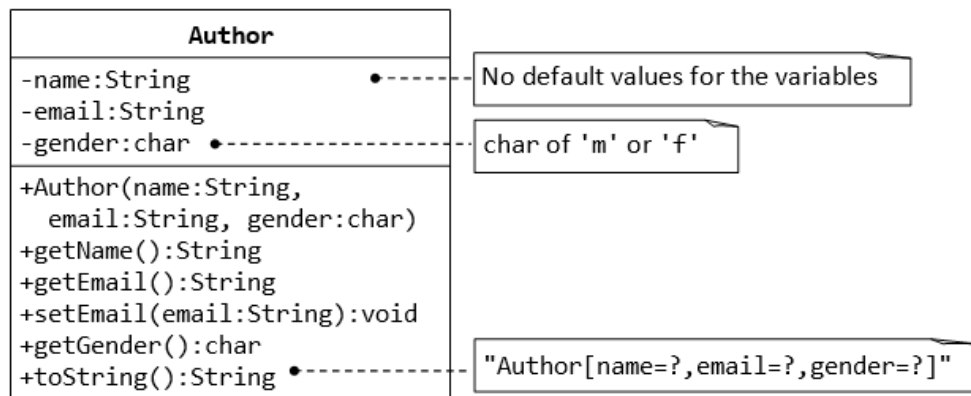
Semester:2nd
Submission deadline 23-02-2020

Subject/Course Learning Outcome	*Taxonomy Level	Ques. Nos.	Marks
Ability to state and explain the basic programming syntax, semantics, building blocks.	L1	4, 5, 6	8
Ability to develop java programs using programming constructs like conditional statements, looping, array, methods and class.	L3	1, 2, 3	7
Ability to analyze, debug and test the programs and correctly predict their outputs.	L2,L3		
Ability to differentiate the behaviours of different data structures and their memory representations.	L4		
Ability to choose appropriate data structures that efficiently model the problem of interest.	L4		
Ability to apply advanced programming techniques for developing solutions of different problems.	L3		

*Blooms taxonomy levels: Knowledge (L1), Comprehension (L2), Application (L3), Analysis (L4), Evaluation (L5), Creation (L6)

Objective: To understand object-oriented principles in java.

Description:



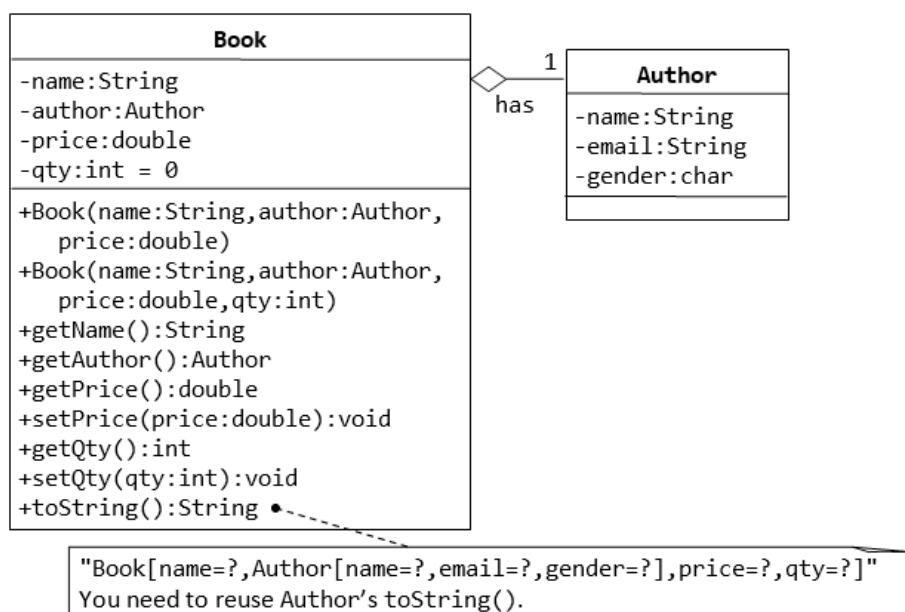
A class called **Author** (as shown in the class diagram) is designed to model a book's author. It contains:

- 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]"`.



A class called **Book** is designed (as shown in the class diagram) to model a book written by *one* author. It contains:

- Four private instance variables: name (String), author (of the class Author you have just created, assume that a book has one and only one author), price (double), and qty (int);
- Two constructors:

```
public Book (String name, Author author, double price) { ..... }  
public Book (String name, Author author, double price, int qty) { ..... }
```

- public methods getName(), getAuthor(), getPrice(), setPrice(), getQty(), setQty().
- A toString() that returns "Book[name=?,Author[name=?,email=?,gender=?],price=?,qty=?".
You should reuse Author's toString().

Create an array of 'n' Book objects to represent books available in a library.

Design a class called **Student** with instance variables: name (String), roll (int), issueDate (Date), returnDate (Date), appropriate constructors and instance methods getName(), getRoll(), issueBook(), depositBook(), and toString().

Date class contains instance variables dd (int), mm (int), yyyy(int), appropriate constructors, and getter methods getDD(), getMM(), getYYYY(), and toString().

issueBook() should first display all the books available in the library with their quantity. A student can issue a book if its quantity is greater than zero, otherwise, it should display a message "Book Not Available" and return NULL. Finally, if the book issued then quantity should be updated, issueDate should be set, and return the Book object. A Student can issue maximum 5 books.

depositBook() should update the quantity of the book returned, set the returnDate, and calculate fine (1Rs. per day) .

Create an array of 'N' students of a class and perform the above operations. Your program should display list of books issued by each student and list of books available in library.

Marks Distribution:

- | | | |
|----|---|-----|
| 1. | Create Author class. | [2] |
| 2. | Create Book class. | [3] |
| 3. | Create Date class. | [2] |
| 4. | Create a library of 'n' books and display. | [2] |
| 5. | Create Student class and implement issueBook(), depositBook() | [4] |
| 6. | Perform the above operations for 'N' students and display complete details. | [2] |

(Create a Tester class to test all the classes and operations. You can add appropriate fields, constructors and methods to Student and Date class if required.)