Java Basics

Training Assignment

Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/05/2019

Hanoi, 05/2020

RECORD OF CHANGES

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	20/May/2020	Createw a new assignment	Create new	DieuNT1	VinhNV

Contents

CODE:	Assignment01_Opt1
TYPE:	Long
LOC:	N/A
DURATION:	90 MINUTES

Java List

Objective

- Java ArrayList, Java Vector, Java Stack

Business needs

- TBD

Working requirements

- Working environment: Eclipse IDE.
- Delivery: Source code, deployment and testing, reviewing evident packaged in a compress archive.

Product architecture

- N/A

Technologies

The product implements one or more technology:

- Java basics
- Java List

Stored Data

- N/A

Exercise 1:

Write a Java program to replace the second element of a ArrayList with the specified element.

Exercise 2:

Write an application that stores the integer numbers between 1 (included) and 64 (included) in an array and prints them in a reverse order.

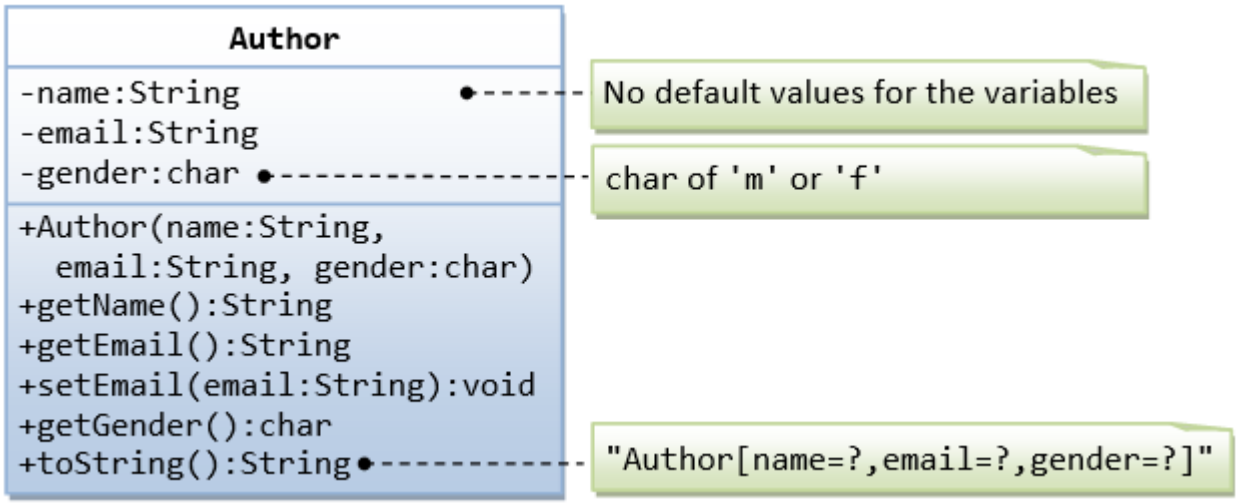
Exercise 3:

Write a Java program to create a Stack data structure. This Stack data structure is to store the integer values. Your program should display a menu of choices to operate the Stack data structure. See the sample menu below:

```
================================================

                Stack  Operations Menu

================================================
1. Add items

2. Delete items

3. Show the number of items

4. Show min and max items

5. Find an item

6. Print all items

7. Exit


Enter your choice:1
```

Exercise 4:

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



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.,

-- THE END --