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

Java OOP(II)

Objective

- Java Inheritance, Java Method Overriding, Java super Keyword, Abstract Class & Method, Java Interfaces, Java Polymorphism, Java Encapsulation

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 OOP (II)

Stored Data

- N/A

Exercise 1:

Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating object of Student class.

Exercise 2:

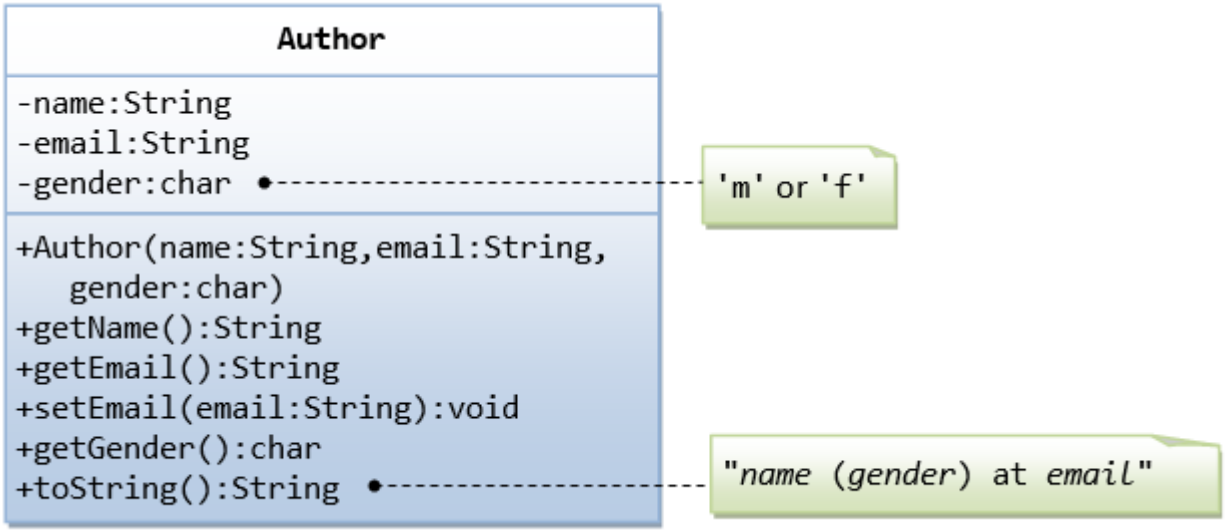
Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

Exercise 3:

Write a class that implements the CharSequence interface found in the java.lang package. Your implementation should return the string backwards. Select one of the sentences from this book to use as the data. Write a small main method to test your class; make sure to call all four methods.

Exercise 4:

Let's start with the Author class



A class called Author is designed as shown in the class diagram. It contains:

Three private member variables: name (String), email (String), and gender (char of either 'm' or 'f' - you might also use a boolean variable called isMale having value of true or false).

A constructor to initialize the name, email and gender with the given values.

(There is no default constructor, as there is no default value for name, email and gender.)

Public getters/setters: getName(), getEmail(), setEmail(), and getGender().

(There are no setters for name and gender, as these properties are not designed to be changed.)

A toString() method that returns "name (gender) at email", e.g., "Tan Ah Teck (m) at ahTeck@somewhere.com".

-- THE END --