



## ***Java SE 8 Programming Language***

# **Lab Guides**


Document Code	25e-BM/HR/HDCV/FSOFT
Version	1.1
Effective Date	20/11/2012

**RECORD OF CHANGES**

No	Effective Date	Change Description	Reason	Reviewer	Approver
1	01/Oct/2018	Add the new labs	Create new	DieuNT1	VinhNV
2	01/Jun/2019	Update	Fsoft template Update Problem Description	DieuNT1	VinhNV

## Contents

Unit 3: Classes and Objects .....	4
Lab Guide 1: Inheritance, Encapsulation.....	4
Objectives:.....	4
Problem Descriptions:.....	4
Functional Requirements: .....	5
Guidelines:.....	5

	CODE:	JPL.S.L201
	TYPE:	SHORT
	LOC:	85
	DURATION:	30 MINUTES

## Unit 3: Classes and Objects

### Lab Guide 1: Inheritance, Encapsulation

#### Objectives: JPL-9

- ✓ Able to create Java-based applications that take advantage of Java object-oriented features, including encapsulation, inheritance, and polymorphism.

#### Problem Descriptions:

Create a new package named **fa.training.entities** in **JPL.S.L201** project that contains:

The **Teacher** abstract class:

- ✓ Instance variables:
  - *designation*: for teacher designation
  - *collegeName*: the collegename that teacher do work
- ✓ Constructor:
  - public **Teacher**(): A default constructor, it should initialize the attribute to null or 0 )
  - public **Teacher** (String designation, String collegeName): A constructor with parameters, it creates the teacher object by setting the two fields to the passed values
- ✓ Instance methods:
  - Getter/Setter methods: are used to get/set the value
  - public void teach(String content){}

The **MathTeacher** class that extends Teacher:

- ✓ Instance variables:
  - *mainSubject*: the main subject
- ✓ Constructor:
  - public **MathTeacher**(): A default constructor, it should initialize the attribute to null or 0 )
  - public **MathTeacher** (String designation, String collegeName, String mainSubject): A constructor with parameters, it creates the teacher object by setting the three fields to the passed values.
- ✓ Instance methods:
  - Getter/Setter methods: are used to get/set the value
  - public void teach(String content){}: override the parent's method
  - public String toString(): This method allows the math teacher to be easily printed out to the screen

Create package **fa.training.management** that contains **TeacherManagement** class:

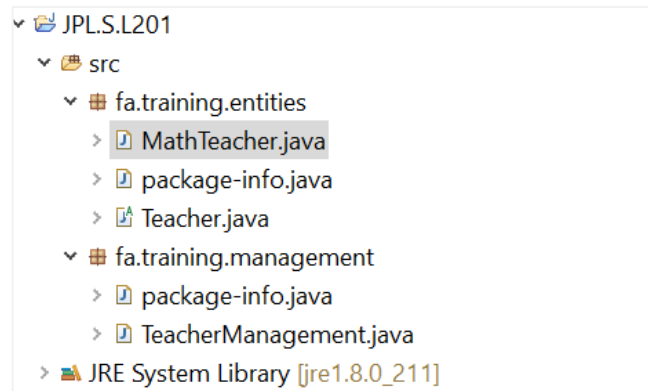
- ✓ Create a new object of MathTeacher.
- ✓ Call method of the class and explains the result.

### **Functional Requirements:**

- a. Change access modifier of attributes of Teacher, MathTeacher. Explain the result and fix.

### **Guidelines:**

Project struture:



- ✓ **Teacher** class

```
1. package fa.training.entities;
2.
3. public abstract class Teacher {
4.     protected String designation;
5.     protected String collegeName;
6.
7.     public Teacher() {
8.     }
9.
10.    public Teacher(String designation, String collegename) {
11.        super();
12.        this.designation = designation;
13.        this.collegeName = collegename;
14.    }
15.
16.    public String getDesignation() {
17.        return designation;
18.    }
19.
20.    public void setDesignation(String designation) {
21.        this.designation = designation;
22.    }
23.
24.    public String getCollegename() {
25.        return collegeName;
26.    }
27.
28.    public void setCollegename(String collegename) {
29.        this.collegeName = collegename;
30.    }
31.
32.    public abstract void teach();
33.
34. }
```

✓ **MathTeacher** class

```
1. package fa.training.entities;
2.
3. /**
4.  *
5.  * @author DieuNT1
6.  *
7.  */
8. public class MathTeacher extends Teacher {
9.     protected String mainSubject;
10.
11.     public MathTeacher() {
12.     }
13.
14.     public MathTeacher(String designation, String collegename,
15.                          String mainSubject) {
16.         super(designation, collegename);
17.         this.mainSubject = mainSubject;
18.     }
19.
20.     public String getMainSubject() {
21.         return mainSubject;
22.     }
23.
24.     public void setMainSubject(String mainSubject) {
25.         this.mainSubject = mainSubject;
26.     }
27.
28.     /**
29.      * The method return sum of all two numbers.
30.      *
31.      * @param number1
32.      * @param number2
33.      * @return an integer value.
34.      */
35.     public int sum(int number1, int number2) {
36.         return (number1 + number2);
37.     }
38.
39.     @Override
40.     public void teach() {
41.         System.out.print("Teaching math subject:");
42.     }
43.
44.     @Override
45.     public String toString() {
46.         return "MathTeacher [mainSubject=" + mainSubject +
47.             ", designation=" + designation + ", collegeName=" +
48.             collegeName + "];"
49.     }
50.
51. }
```

✓ **TeacherManagement** class

```
1. package fa.training.management;
2.
3. import fa.training.entities.MathTeacher;
4.
5. public class TeacherManagement {
6.
7.     public static void main(String[] args) {
8.
9.         MathTeacher teacher = new MathTeacher("Teacher", "FU", "Math");
10.        System.out.println(teacher);
11.
12.        teacher.teach();
13.        System.out.println(" sum(20, 80) = " + teacher.sum(20, 80));
14.
15.    }
16.
17. }
```

## ✓ How to run:

Click **Run** menu | choose **Run as**:

**Results:**

```
MathTeacher [mainSubject=Math, designation=Teacher, collegeName=FU]
Teaching math subject: sum(20, 80) = 100
```

-- THE END --