

Exercise 01:

Recall the following scenario discussed during the class. Develop a code base to represent the scenario. Add a test class to invoke Lecturer and Student class by creating atleast one object from each.

Note: All the common attributes and behavior stored in the super class and only the specific fields and behavior stored in subclasses.

Student	Lecturer	Person
- name	- name	Identify field and attributes to be stored in this class
- id	- id	
- course	- programme	
+ setName()/getName()	+ setName()/getName()	
+ setID()/getID()	+ setID()/getID()	
+ setCourse()/getCourse()	+ setProg()/getProg()	

Answers

//person class

```
public class Person {  
  
    private String name;  
  
    private int id;  
  
    // Getters and Setters for name and id  
  
    public String getName() {  
  
        return name;  
  
    }  
  
    public void setName(String name) {  
  
        this.name = name;  
  
    }  
  
}
```

```
public int getID() {  
    return id;  
}  
  
public void setID(int id) {  
    this.id = id;  
}  
}
```

//Student class

```
public class Student extends Person {  
    private String course;  
  
    // Getter and Setter for course  
    public String getCourse() {  
        return course;  
    }  
  
    public void setCourse(String course) {  
        this.course = course;  
    }  
}
```

//lecturer

```
public class Lecturer extends Person {  
    private String programme;
```

```
// Getter and Setter for programme

public String getProgramme() {

    return programme;

}

public void setProgramme(String programme) {

    this.programme = programme;

}

}
```

//test person class (student and lecturer)

```
public class TestPerson {

    public static void main(String[] args) {

        // Create a Student object

        Student student = new Student();

        student.setName("Mohamed rila");

        student.setID(27868);

        student.setCourse("Software engineering");


        // Create a Lecturer object

        Lecturer lecturer = new Lecturer();

        lecturer.setName("Mr.shafraz");

        lecturer.setID(1234);

        lecturer.setProgramme("Java devoloper");

    }

}
```

```
// Print student details

System.out.println("Student Name: " + student.getName());

System.out.println("Student ID: " + student.getID());

System.out.println("Student Course: " + student.getCourse());


// Print lecturer details

System.out.println("\nLecturer Name: " + lecturer.getName());

System.out.println("Lecturer ID: " + lecturer.getID());

System.out.println("Lecturer Programme: " + lecturer.getProgramme());

}

}
```

Exercise 02

Develop the following class execute and discuss the answer: Please note that each public class stored in separate files. Write down the answer.

```
public class Animal{}

public class Mammal extends Animal{}

public class Reptile extends Animal{}


public class Dog extends Mammal{

    public static void main(String args[]){

        Animal a = new Animal();
```

Practical 05: Encapsulation & Inheritance

```
Mammal m = new Mammal();  
  
Dog d = new Dog();  
  
System.out.println(m instanceof Animal);  
  
System.out.println(d instanceof Mammal);  
  
System.out.println(d instanceof Animal);  
  
}  
}
```

Answers

1. We have four classes: **'Animal'**, **'Mammal'**, **'Reptile'**, and **'Dog'**. Each class is stored in separate files.

2. **'Animal'** class:

- It is a superclass that doesn't have any specific fields or methods in this example.

3. **'Mammal'** class:

- It extends the **'Animal'** class, which means it is a subclass of **'Animal'**.
- As **'Mammal'** is a subclass of **'Animal'**, it inherits all the members (fields and methods) of **'Animal'**.

4. **'Reptile'** class:

- It also extends the **'Animal'** class and inherits its members.

5. **'Dog'** class:

- It extends the **'Mammal'** class, which means it is a subclass of both **'Mammal'** and **'Animal'**.
- As a subclass of **'Mammal'**, it inherits all the members (fields and methods) of **'Mammal'**, including those inherited from **'Animal'**.

6. In the **'main'** method of **'Dog'** class, we create three objects of different classes: **'Animal'**, **'Mammal'**, and **'Dog'**.

7. We then use the **'instanceof'** operator to check whether each object is an instance of a particular class:

- **'m instanceof Animal'**: As **'m'** is an object of **'Mammal'** class, which extends **'Animal'**, this will be true.
- **'d instanceof Mammal'**: As **'d'** is an object of **'Dog'** class, which extends **'Mammal'**, this will also be true.
- **'d instanceof Animal'**: As **'d'** is an object of **'Dog'** class, which extends **'Mammal'**, and **'Mammal'** extends **'Animal'**, this will be true as well