

Practical 04: Encapsulation & Inheritance

Exercise 01:

Create a class called "Employee" which has 3 private variables (empID, empName, empDesignation) and create getters and setters for each field. Please note that this has no main method since this is just a blueprint not a application. Now crate a test class to invoke the Employee class. Create two objects for Mr.Bogdan and Ms.Bird and set required values using setters and print them back on the console using getters.

Exercise 02:

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

```
class SuperB {  
  
    int x;  
  
    void setIt (int n) { x=n;}  
  
    void increase () { x=x+1;}  
  
    void triple () {x=x*3;};  
  
    int returnIt () {return x;}  
  
}  
  
class SubC extends SuperB {  
  
    void triple () {x=x+3;} // override existing method  
  
    void quadruple () {x=x*4;} // new method  
  
}  
  
public class TestInheritance {  
  
    public static void main(String[] args) {  
  
        SuperB b = new SuperB();  
  
        b.setIt(2);  
  
        b.increase();  
  
        b.triple();  
  
    }  
}
```

Practical 04: Encapsulation & Inheritance

```
        System.out.println( b.returnIt() );

        SubC c = new SubC();

        c.setIt(2);

        c.increase();

        c.triple();

        System.out.println( c.returnIt() ); }

}
```

Exercise 03:

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()	

Exercise 04

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{}
```

Practical 04: Encapsulation & Inheritance

```
public class Dog extends Mammal{  
  
    public static void main(String args[]){  
  
        Animal a = new Animal();  
  
        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);  
  
    }  
}
```