**TOPIC : DESIGN PATTERNS AND PRINCIPLES**

EXERCISE 2: IMPLEMENTING THE FACTORY METHOD PATTERN

public interface Animal {

void speak();

}

public class Cat implements Animal{

public void speak() {

System.***out***.println("cat meows");

}

}

public class Dog implements Animal{

public void speak() {

System.***out***.println("dog barks");

}

}

public class Petshop {

public Animal getAnimal(String type) {

if (type == null) {

return null;

}

if (type.equalsIgnoreCase("DOG")) {

return new Dog();

} else if (type.equalsIgnoreCase("CAT")) {

return new Cat();

} else {

System.***out***.println("No Animal");

return null;

}

}

}

public class TestFactory {

public static void main(String[] args) {

Petshop factory = new Petshop();

Animal a1 = factory.getAnimal("Dog");

if (a1 != null) a1.speak();

Animal a2 = factory.getAnimal("Cat");

if (a2 != null) a2.speak();

Animal a3 = factory.getAnimal("Lion");

if (a3 != null) a3.speak();

}

}

**OUTPUT SCREENSHOT:**

