

PGDIT 2018
[OOP Term Paper (Homework)]

1. Design a class for the following scenario:

Suppose you want to create several **Animal** objects and assign each a **serialNumber**, beginning with 1001. Once the ID is assigned to an object, it never changes.

At the same time, you need a variable **numberOfAnimals** to keep track of how many **Animal** objects are created and this field (variable) is not related to any individual object that means a class variable.

You cannot introduce any new variable other than **serialNumber** and **numberOfAnimals**. In your class, there should at least be static, final, private, public keywords and also getter and setter methods.

2. Create an **Exception** class named *InvalidDigitException*; an object of this class is thrown by the method *setDigit(digit: int)* of PhoneNumber Class when the parameter of the method is not a digit (0 to 9). Define the method *setDigit(digit: int)* in two different ways for the Class PhoneNumber whose UML diagrams is given below. Search on the net about the UML diagram of a class.

Way 1:

PhoneNumber
-digit: int
+PhoneNumber(digit: int) +setDigit(digit:int): void throws InvalidDigitException +toString(): String

Way 2:

PhoneNumber
-digit: int
+PhoneNumber(digit: int) +setDigit(digit:int): void +toString(): String

Please note that the only difference in two ways is how the method *setDigit()* handles the exception. In the first case, the method should not handle the exception, but the method that called the *setDigit()* should handle the exception with try/catch/finally where necessary. In the second case, the *setDigit()* method itself should handle the exception and surround the code with try/catch/finally where the exception is thrown.

3. Write a java code that reads every lines from a file named 'input.txt' and write the line number followed by the number of words in that line into the file output.txt. Also, add try, catch and finally blocks where necessary.

Inside the input.txt file, sample input:

This is my country.
The capital of my country is Dhaka.
It's a beautiful city.

Inside the output.txt file, sample Output:

1 4
2 7
3 4

- 4 In the Mobile system of our country, service can be divided into two broad categories based on the billing methods. First type is known as **Prepaid** service and it charges a fixed monthly bill at the beginning of each month. The total bill is calculated by summing up the fixed bill and the vat on the fixed bill. Another type is known as **Postpaid** service and the total bill is calculated by multiplying the total minutes of talk times and rate per minute, plus the vat.

Design a class MobilePackage with its attributes and methods including an abstract method calculateTotalBill() which calculates the bill of each service and a protected attribute (variable) totalBill which keeps the total bill amount. Design two classes PrepaidPackage and PostpaidPackage that inherit the MobilePackage abstract class.

Sub-classes have to implement the abstract method calculateTotalBill(). The toString() method is overridden in the two sub-classes to print the operator name and total bill amount. You need to add constructor methods where you think necessary. The class descriptions are given in the UML diagram (search net) below and note that constructor methods are not shown in the diagram.

