H HackerRank
For Work

Question - 1
Animal Inheritance   SCORE: **50 points**

General Programming   Inheritance   Polymorphism   Easy   Abstraction   Java   C++   C#
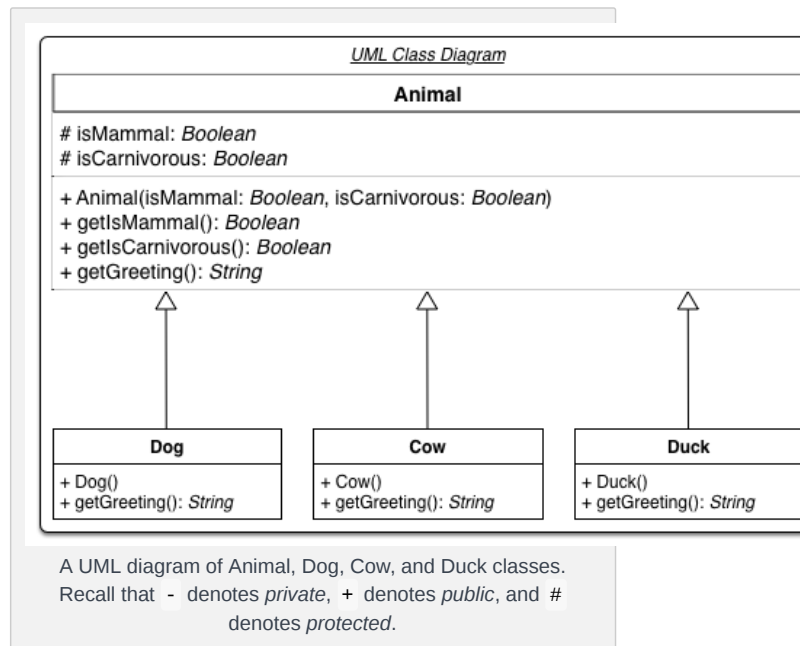
Object Oriented Programming   Java 8   Core Skills   Problem Solving

In this challenge, you will be asked to build on an abstract class and
initialize an instance of each class with a variable.   The program will then
test your implementation by retrieving the data you stored.

The locked code in the editor does the following:
1. Declares an abstract class named `Animal` with the
   implementations for `getIsMammal()` and
   `getIsCarnivorous()` methods, as well as an abstract method
   named `getGreeting()`.
2. Creates *Dog*, *Cow*, and *Duck* objects.
3. Calls the `getIsMammal()`, `getIsCarnivorous()`, and
   `getGreeting()` methods on each of these respective objects.

Consider the following UML diagram:

```
                          UML Class Diagram
  ┌───────────────────────────────────────────────────────────┐
  │                          Animal                            │
  ├───────────────────────────────────────────────────────────┤
  │ # isMammal: Boolean                                        │
  │ # isCarnivorous: Boolean                                   │
  ├───────────────────────────────────────────────────────────┤
  │ + Animal(isMammal: Boolean, isCarnivorous: Boolean)        │
  │ + getIsMammal(): Boolean                                   │
  │ + getIsCarnivorous(): Boolean                              │
  │ + getGreeting(): String                                    │
  └───────────────────────────────────────────────────────────┘
          △                    △                    △

  ┌─────────────────┐  ┌─────────────────┐  ┌─────────────────┐
  │      Dog        │  │      Cow        │  │      Duck       │
  ├─────────────────┤  ├─────────────────┤  ├─────────────────┤
  │ + Dog()         │  │ + Cow()         │  │ + Duck()        │
  │ + getGreeting():│  │ + getGreeting():│  │ + getGreeting():│
  │   String        │  │   String        │  │   String        │
  └─────────────────┘  └─────────────────┘  └─────────────────┘
```

A UML diagram of Animal, Dog, Cow, and Duck classes.
Recall that `-` denotes *private*, `+` denotes *public*, and `#`
denotes *protected*.

Complete the code in the editor below to implement the following:
1. Three classes named `Dog`, `Cow`, and `Duck` that inherit from the
   `Animal` class.
2. No-argument constructors for each class that initialize the instance
   variables inherited from the superclass.
3. Each class must implement the `getGreeting()` method:
   - For a *Dog* object, this must return the string `ruff`.
   - For a *Cow* object, this must return the string `moo`.
   - For a *Duck* object, this must return the string `quack`.

Input Format
There is no input for this challenge.

Output Format

The `getGreeting()` method must always return a string denoting the
appropriate greeting for the implementing class.

Sample Output

```
A dog says 'ruff', is carnivorous, and is a
mammal.
A cow says 'moo', is not carnivorous, and is a
mammal.
A duck says 'quack', is not carnivorous, and is
not a mammal.
```

Question - 2

SCORE: **20 points**

**What happens when you try to compile and run the following
program?**

```
public class Demo {
    static int x = 0;

    Demo() {
        x ++ ;
    }
    void get(){
        System.out.print("-x" + x);
    }
    public static void main(String[] args){
        Demo d1,d2,d3,d4;
        Demo d5 = new Demo();
        Demo d6 = new Demo();
        Demo d7 = new Demo();
        d7.get();
    }
}
```

- ⚪ This program is illegal at runtime time.

- ⚪ This program writes nothing to the standard output.

- ⚪ This program writes "-x0" to the standard output.

- ⚪ This program writes "-x1" to the standard output.

- ⦿ This program writes "-x3" to the standard output.

- ⚪ This program writes "-x4" to the standard output.

- ⚪ This program writes "-x6" to the standard output.

- ⚪ This program writes "-x7" to the standard output.

Question - 3

SCORE: **20 points**

Which answer(s) match the given pattern?

Which answer(s) match the given pattern?
pattern: a*b+

- ⚪ aaaaa

- ⦿ aaaaab

- ⦿ b
- ◯ a
- ◯ aaa
- ⦿ abbbb
- ⦿ aaabbb

- ⦿ b
- ◯ a