Exercise - Polymorphism

Figure 1: Kitten

For this exercise, you will write three classes:

- `Animal.java`
- `Centipede.java`
- `Cat.java`

We have provided you with `Runner.java` main class to help you write your classes. Take a look at it before you start and uncomment the lines in it as you make your way through the assignment.

Animal - Superclass

First, you will need to implement the `Animal` class. This class must have three *private* fields:

- `name` [String, initialized to "Animal"]
- `noise` [String, not initialized]
- `numLegs` [int, not initialized]

Write a constructor that takes a single integer and uses it to set the value of the `numLegs` variable.

Write *public* getters and setters for each of these fields. Look at `Runner.java` to see how these methods should be named.

Once you have finished the `Animal` class, you should get the output below when you run it:

```
Class: class Animal
Name: Animal
Number of legs: 4
Noise: null
```

Centipede - Subclass

Next up, write the `Centipede` class. This class must inherit from the `Animal` class and has no extra fields.

Write a constructor that takes no arguments. It should call the constructor of its parent class with the value 100. It should also set the `name` field of the parent class to the string "Centipede".

Override the `getNoise` method so that it returns the string `‘‘Centipedes don’t make noise.’’`

Once you have finished, the `Runner` class should output the following:

```
Class: class Animal
Name: Animal
Number of legs: 4
Noise: null

Class: class Centipede
Name: Centipede
Number of legs: 100
Noise: Centipedes don’t make noise.

Class: class Centipede
Name: Centipede
Number of legs: 100
Noise: Centipedes don’t make noise.
```

Cat - Subclass

Write the subclass `Cat`. This class must inherit from `Animal` and has no extra fields.

Write a constructor that does the following:

- Calls the constructor of the parent class with the argument 4
- Uses the parent setter to set the `name` field to "Cat".
- Uses the parent setter to set the `noise` field to "Meow".

Override the `Object toString` method so that it returns the string `‘‘An adorable kitten.’’`.

If you have written the `Cat` class correctly, the output of running `Runner.java` should look like the following:

```
Class: class Animal
Name: Animal
Number of legs: 4
Noise: null

Class: class Centipede
Name: Centipede
Number of legs: 100
Noise: Centipedes don't make noise.

Class: class Centipede
Name: Centipede
Number of legs: 100
Noise: Centipedes don't make noise.

Class: class Cat
Name: Cat
Number of legs: 4
Noise: Meow
An adorable kitten.

Class: class Cat
Name: Cat
Number of legs: 4
Noise: Meow
An adorable kitten.
```

Submit your `Animal.java`, `Centipede.java`, and `Cat.java` files in a zip file to the Autograder.