10.5 Exercises

Factual

Recall that the `maxDog` method has the following signature:

```
public static Dog maxDog(Dog d1, Dog d2) { … }
```

1. What is the static type of `Dog.maxDog(dogC, dogD)`?

```
ShowDog dogC = new ShowDog("Franklin", "Malamute", 180, 6);
ShowDog dogD = new ShowDog("Gargamel", "Corgi", 44, 12);

Dog.maxDog(dogC, dogD);
```

2. Which (if any), will compile:

```
Dog md = Dog.maxDog(dogC, dogD);
ShowDog msd = Dog.maxDog(dogC, dogD);
```

3. In the code below, what are the dynamic types of `o`, `d`, `stuff[0]`, and `stuff[1]`?

```
Object o = new Dog("Hammy", "Beagle", 15);
Dog d = new ShowDog("Ammo", "Labrador", 54);
Object stuff[] = new Object[5];
stuff[0] = o;
stuff[1] = d;
studd[2] = null;
```

> Problem 1

> Problem 2

> Problem 3

Conceptual

1. Is it possible for an interface to extend a class? Provide an argument as to why or why not.
2. What are the differences between `extends` and `implements` inheritance? Is there a particular time when you would want to use one over the other?

> Problem 1

> Problem 2

Procedural

1. Say there is a class `Poodle` that inherits from `Dog`. The Dog class looks like this:

```
public class Dog {
  int weight;
  public Dog(int weight_in_pounds) {
    weight = weight_in_pounds;
  }
}
```

And the Poodle class looks like this:

```
public class Poodle extends Dog {
 public Poodle() {}
}
```

Is this valid? If so, explain why. If it is not valid, then explain how we can make it valid.

2. The `Monkey` class is a subclass of the `Animal` class and the `Dog` class is a subclass of the `Animal` class. However, a Dog is not a Monkey nor is a Monkey a Dog. What will happen for the following code? Assume that the constructors are all formatted properly.

```
Monkey jimmy = new Monkey("Jimmy");
Dog limmy = (Dog) jimmy;
```

3. How about for this code? Provide brief explanation as to why you believe your answers to be correct.

```
Monkey orangutan = new Monkey("fruitful");
Dog mangotan = ((Dog) ((Animal) orangutan));
```

> Problem 1

> Problem 2

> Problem 3

Metacognitive

1. Problem 1 from the Spring 2018 Midterm 1
2. Problem 1 from the Spring 2017 Midterm 1

> Problem 1

> Problem 2

Previous
10.4 Higher Order Functions in Java

Next
11. Inheritance III: Subtype Polymorphism, Comparators, Comparable

Last updated 7 months ago