CS 61BL Summer 2021

Inheritance

Quiz 4: Wednesday July 7, 2021

Note that this quiz is unlike usual quizzes in that you get instant feedback if your answer is correct, and you can retry questions as many times as you like! You can also watch this video for a walkthrough of the solutions.

1 Constructors

We define the following classes:

```
1  public class Animal {
2
3      private String sound;
4      private int numLegs;
5      private boolean domesticated;
6
7      public Animal(String s, int n, boolean d) {
8          // implementation not shown
9      }
10
11     public Animal(Animal animal) {
12         // implementation not shown
13     }
14 }
15 public class Cat extends Animal {
16     public Cat(String s, int n, boolean d) {
17         // Line 1 goes here
18     }
19     public Cat(Cat cat) {
20         // Line 2 goes here
21     }
22 }
```

Part 1

In order to finish the implementation for Cat, which of the following lines are valid replacements for the line `// Line 1 goes here`? We say a replacement is valid if the resulting code compiles. Consider each line independently (i.e. selecting a choice means that you are saying that line alone will be what replaces the line `// Line 1 goes here`).

[ ] `super()`
[ ] `super(s, n, d)`
[ ] `super(new Animal())`
[ ] `super(null)`
[ ] No line is needed to be valid

Part 2

In order to finish the implementation for Cat, which of the following lines are valid replacements for the line `// Line 2 goes here`? We say a replacement is valid if the resulting code compiles. Consider each line independently (i.e. selecting a choice means that you are saying that line alone will be what replaces the line `// Line 2 goes here`).

[ ] `super()`
[ ] `super(s, n, d)`
[ ] `super(new Animal())`
[ ] `super(null)`
[ ] No line is needed to be valid

2 Inheritance

Suppose we have the classes `A`, `B`, and `C` below:

```
class A {
    public void go(B other) {
        System.out.println("bears");
    }

    public void go(A other) {
        System.out.println("cats");
    }
}

class B extends A {
    public void go(A other) {
        System.out.println("ravens");
    }
}

class C extends A {
    public void go(C other) {
        System.out.println("eagles");
    }
}
```

Suppose we also have the following objects:

```
A sohum = new B();
A allyson = new A();
C zoe = new C();
```

For each function call below, write what is printed

```
allyson.go(allyson);
```

( ) bears ( ) cats ( ) ravens ( ) eagles

```
allyson.go(zoe);
```

( ) bears ( ) cats ( ) ravens ( ) eagles

```
sohum.go(allyson);
```

( ) bears ( ) cats ( ) ravens ( ) eagles

```
zoe.go(sohum);
```

( ) bears ( ) cats ( ) ravens ( ) eagles

```
zoe.go(zoe);
```

( ) bears ( ) cats ( ) ravens ( ) eagles

3 Casting

Suppose we have the same classes from the previous part, reiterated below:

```
class A {
    public void go(B other) {
        System.out.println("bears");
    }

    public void go(A other) {
        System.out.println("cats");
    }
}

class B extends A {
    public void go(A other) {
        System.out.println("ravens");
    }
}

class C extends A {
    public void go(C other) {
        System.out.println("eagles");
    }
}
```

And the same instances, reiterated below:

```
A sohum = new B();
A allyson = new A();
C zoe = new C();
```

For each line of code below, determine if it runs successfully. If it doesn't, determine whether a compilation error or a runtime error occurs.

```
A one = (A) sohum;
```

( ) Runs successfully ( ) Compilation Error ( ) Runtime Error

```
B two = (B) sohum;
```

( ) Runs successfully ( ) Compilation Error ( ) Runtime Error

```
A three = (A) zoe;
```

( ) Runs successfully ( ) Compilation Error ( ) Runtime Error

```
C four = (C) sohum;
```

( ) Runs successfully ( ) Compilation Error ( ) Runtime Error

```
B five = (B) zoe;
```

( ) Runs successfully ( ) Compilation Error ( ) Runtime Error