Does this code compile?

Assuming that `GuardDog` inherits from `Dog`, that its constructor takes a `String` and an `int`, and that it has a `growl` method, does the following code compile? Why or why not?

```java
Dog d = new GuardDog("fluffy", 1);
d.growl();
```
**interface**
*what* a piece of code can do

**implementation**
*how* a piece of code works

**type**
describe a set of supported operations

**class**
implement a type’s operations

**subtype**
add more operations to an existing type

**subclass**
re-use/modify an existing implementation

**inheritance**
usually extends interface *and* implementation
Declared type

When we declare a variable to be of a particular type, the value of that variable must always be an instance of that type.

If a variable has a type, then the value of that variable can be used anywhere that type is expected.

```java
void f(int x) {
    ...
}
...
int value = 3;
f(value);
int y = value;

void g(Dog d) {
    ...
}
...
Dog buddy = new Dog(...);
g(buddy);
Dog myDog = buddy;
```
Subtyping: the “is-a” relationship

Implementing an interface establishes an is-a relationship.

Extending an interface establishes an is-a relationship.

Extending a class establishes an is-a relationship.

If we have the following declaration

```
Type variable;
```

then:

- variable’s declared type is Type.
- variable “is a” Type.
- If Type is an interface, then variable “is” all the interfaces that Type transitively extends.
- If Type is a class, then variable “is” all the classes that Type transitively extends \textit{and} “is” all the interfaces that Type transitively implements.
Subtyping as substitutability

When we declare a variable to be of a particular type, we say that the value of that variable should always be an instance of that type or one of its subtypes. If a variable has a type, then the value of that variable can be used anywhere that type or one of its supertypes is expected.

```java
void g(Animal A) {
    ...
}

...

Dog buddy = new Dog(...);
g(buddy);
Animal myDog = buddy;
```
Subtyping as substitutability

When we declare a variable to be of a particular type, we say that the value of that variable should always be an instance of that type or one of its subtypes.

If a variable has a type, then the value of that variable can be used anywhere that type or one of its supertypes is expected.

```java
void g(Animal A) {
  ...
}

Dog buddy = new Dog(...);
g(buddy);
Animal myDog = buddy;
```
Subtyping as substitutability

When we declare a variable to be of a particular type, we say that the value of that variable should always be an instance of that type or one of its subtypes.

If a variable has a type, then the value of that variable can be used anywhere that type or one of its supertypes is expected.

```java
void g(Animal A) {
    ...
}

...

Dog buddy = new Dog(...);
g(buddy);  // is-a
Animal myDog = buddy;  // subtype
```
Declared type vs actual type

The type checker looks at the declared type (not the value) to see if method calls are legal.

\[
x.\text{getSpots}() \text{ is legal only if the declared type of } x \text{ guarantees there's a getSpots method.}
\]

When code runs, Java looks at the actual object (not the claimed type) to choose the right method to run.

\[
\text{animal.speak()} \text{ does different things, depending on what kind of object } \text{animal} \text{ is currently referencing.}
\]
Inheritance Puzzles

Does this type check in Java?

Cat c = new Cat("Nala", 14);
Cat c = new Cat("Nala", 14);
c.speak();

Meow
Inheritance Puzzles

Does this type check in Java?

Animal a = new Cat("Nala", 14);
Inheritance Puzzles

Does this type check in Java?

Animal a = new Cat("Nala", 14);
a.speak();

✔ Meow
Inheritance Puzzles

Does this type check in Java?

Dog d = new Dalmatian("Pango", 3, 101);
Does this type check in Java?

Dalmatian d = new Dog("Pango", 101);
Inheritance Puzzles

Does this type check in Java?

Dog d = new Dalmatian("Pango", 3, 101);
Dalmatian dm = d;
Inheritance Puzzles

Does this type check in Java?

GuardDog gd = new GuardDog("fluffy", 1);
gd.growl();

✔
Inheritance Puzzles

Does this type check in Java?

Dog d = new GuardDog("fluffy", 1);
d.growl();
Sydney 2000
Syd.
tennis

Courtesy of Prof. Bassman