

COMP 250

Assignment Project Exam Help

INTRODUCTORY SCIENCE

<https://eduassistpro.github.io/>

Week 4-5: COLLECT

Add WeChat edu_assist_pro

Giulia Alberini, Fall 2020

WHAT ARE WE GOING TO DO IN THIS VIDEO?



OOD6

Assignment Project Exam Help

- Modifiers and Inher
- Type Conversion

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

MODIFIERS

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

ACCESS CONTROL MODIFIERS

- Recall that a class can be declared to be either `public` or `package-private` (no keyword).
- A class can *extend* another class. The latter is visible from where the former is located.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

ACCESS CONTROL MODIFIERS

- Recall that a class can be declared to be either `public` or package-private (no keyword).
- A class can *extend* another class. The latter is visible from where the former is located.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
package assignments.a1;  
  
public class A {  
    :  
}
```

```
package assignments;  
import assignments.a1.A;  
  
public class B extends A{  
    :  
}
```



All public classes
can be extended
(even across
packages)

ACCESS CONTROL MODIFIERS

- Recall that a class can be declared to be either `public` or package-private (no keyword).
- A class can *extend* another class if the latter is visible from where the former is located.

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
package assignments.a1;  
  
class A {  
    :  
}
```

```
package assignments;  
  
public class B extends A {  
    :  
}
```



Not allowed,
since A is not
visible from B.

WHICH MEMBERS ARE INHERITED?

- Every superclass' member visible from where the subclass is located is inherited by the subclass (with the exception of constructors)

Assignment Project Exam Help

<https://eduassistpro.github.io/>

- Members include: fi

c nested cla .

Add WeChat edu_assist_pro

- Note that *a subclass cannot reduce the visibility of an inherited method*. The visibility can only be increased. (we'll understand better why in the next few classes)

ASIDE: NESTED CLASSES

- Note that a nested class is not a subclass.

Assignment Project Exam Help

- Outer and inner classes and methods of each other. Details are out of the scope of

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

`final` KEYWORD

- A class that has been declared `final` cannot be *extended*.

Assignment Project Exam Help

```
public final class Dog {  
    :  
}
```

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
public class eagle extends Dog {  
    :  
}
```



compile-time error!

`final` KEYWORD

- A method that has been declared `final` cannot be *overridden*.

Assignment Project Exam Help

```
public class Dog {  
    public final void bark() {  
        :  
    }  
}
```

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
ss Beagle extends Dog {  
    id bark() {  
    }  
}
```



compile-time error!

Assignment Project Exam Help
TY ON
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro



FROM LAST A COUPLE OF VIDEOS AGO

class Dog

Person owner

```
public void bark() {  
    print("woof!");  
}
```

:

↑ extends

class Beagle

void hunt ()

```
public void bark() {  
    print("aowwwuuu");  
}
```

:

```
public class Test {  
    public static void main(String[] args) {  
        log.silly(new Beagle());  
    }  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Is this
allowed??

OBJECTS TYPE

- We have seen that an object is of the type of the class from which it was instantiated.

Assignment Project Exam Help

- For example, if we write <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

```
Dog myDog = new Dog();
```

then `myDog` points to an object of type `Dog`.

OBJECT TYPES

- But Dog is a subclass of Animal which is a subclass of Object.

Assignment Project Exam Help

- Thus, a Dog is an Animal type Dog wherever object are called for.

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

- Note that the reverse is not necessarily true: an Animal could be a Dog, but not necessarily. Similarly, an Object could be an Animal or a Dog, but it isn't necessarily.

TYPE CASTING – REFERENCE TYPES

- Casting allows us to use an object of one type in place of another type, if permitted.

Assignment Project Exam Help

- For example we can write <https://eduassistpro.github.io/>

Add WeChat edu_assist_pro
`Animal myPet = new Dog();`

This will not cause a compile-time error because there is an ***implicit upcasting*** since a `Dog` is for sure also an `Animal`.

TYPE CASTING – REFERENCE TYPES

On the other hand, consider the following

```
Animal myPet = new Dog();  
Dog myDog = myPet;
```

<https://eduassistpro.github.io/>

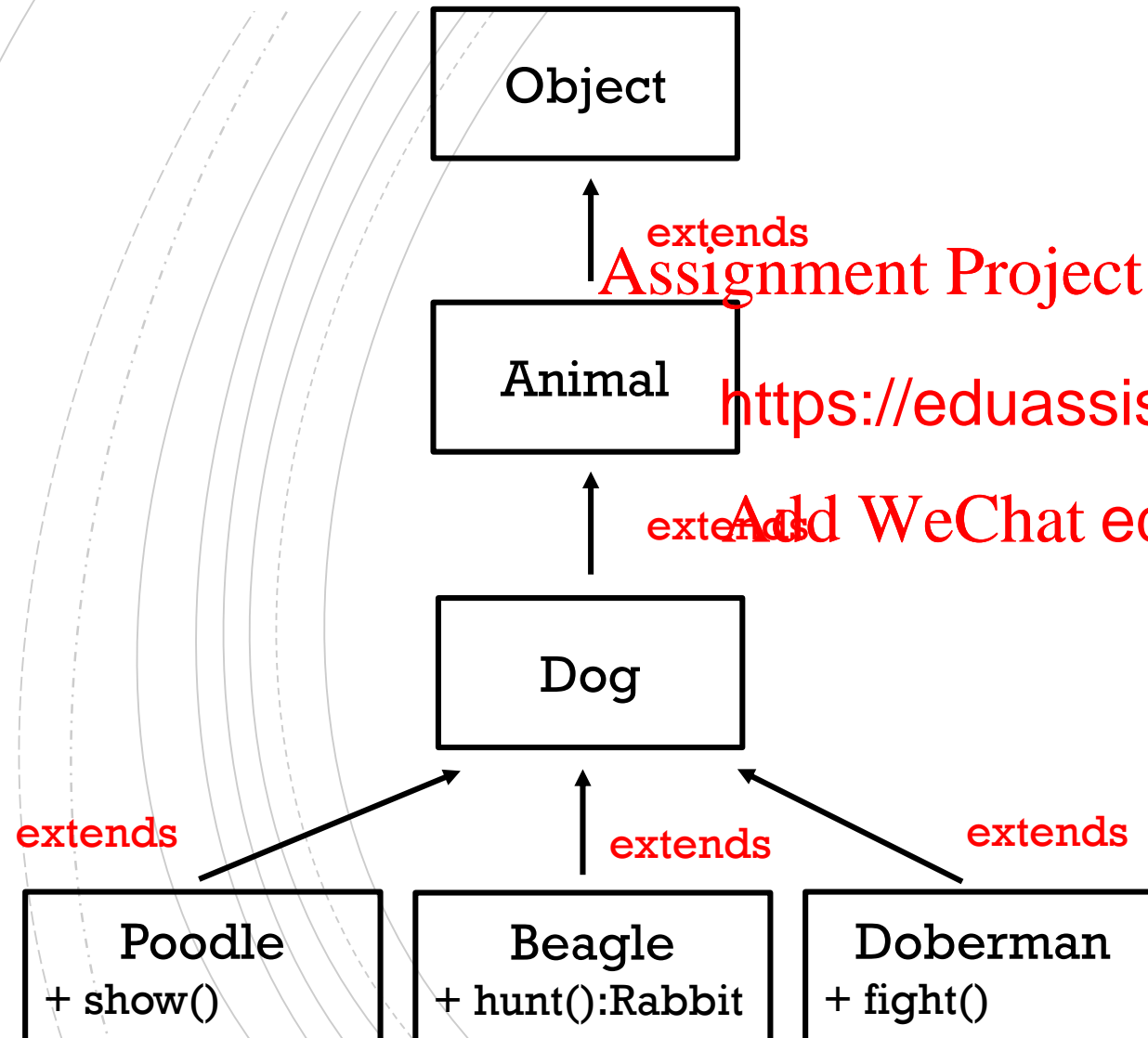
The second line will cause a compile-time error. From the compiler point of view, `myPet` is of type `Animal` and an `Animal` is not a `Dog`.

However, we can tell the compiler that `myPet` is of the correct type, by ***explicitly downcasting***:

```
Dog myDog = (Dog) myPet;
```

If `myPet` turns out to be of the wrong type we'll get a run-time error.

HIERARCHY FROM LAST CLASS



Upcasting

Happens automatically

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

IMPORTANT!

Note that casting does NOT change the object itself, it just labels it differently!

Downcasting

The programmer has to manually do it.

EXAMPLES

`Dog myDog = new Beagle();`

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Is this allowed?

- Yes, it is an example of upcasting which happens automatically.

EXAMPLES

```
Dog myDog = new Beagle();  
Poodle
```

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

Is this allowed?

- **Compile-time error!** The variable `myDog` is of type `Dog`, and it might not be pointing to a `Poodle`. It requires explicit downcasting to compile.

EXAMPLES

```
Dog myDog = new Beagle();  
Poodle myDog;
```

<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

Is this allowed?

- The code compiles, but there will be a **run-time error** because `myDog` is not pointing to a `Poodle` after all.

EXAMPLES

```
Dog myDog = new Beagle();  
myDog.hunt();
```

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

Is this allowed?

- **Compile-time error!** The variable `myDog` is of type `Dog`, and there is no method called `hunt` inside the `Dog` class.

EXAMPLES

```
Dog myDog = new Beagle();  
( (Beagle)
```

Assignment Project Exam Help
<https://eduassistpro.github.io/>
Add WeChat edu_assist_pro

Is this allowed?

➤ Yes, this code will compile and run.



NEXT VIDEO!

class Dog

Person owner

```
public void bark() {  
    print("woof!");  
}
```

:

↑ extends

class Beagle

void hunt ()

```
public void bark() {  
    print("aowwwuuu");  
}
```

:

```
public class Test {  
    public static void main(String[] args) {  
        log.silly(new Beagle());  
    }  
}
```

Assignment Project Exam Help

<https://eduassistpro.github.io/>

Add WeChat edu_assist_pro

Is this
allowed??

Which
bark() will
execute???

Yes, it's an
example of
upcasting!



Coming Soon

Assignment Project Exam Help

In the next

- Polymor <https://eduassistpro.github.io/>
Add WeChat edu_assist_pro