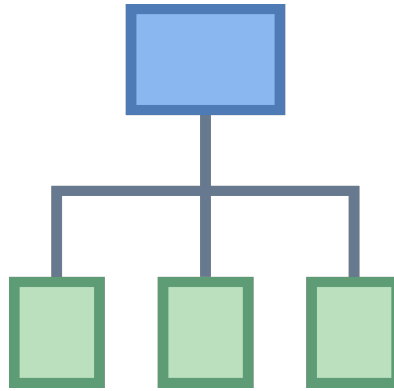


How to Make an App For Beginners

Module 2
Lesson 3
Worksheet



Welcome!

This worksheet will help you practice and understand class inheritance and subclassing.

Follow the instructions below and do it in a playground on your own computer.

Just in case you don't remember how you got here (or if you need a refresher), Lesson 3 can be found here:

<https://www.youtube.com/watch?v=fCyM525RM48>

Step #1: We'll be doing these exercises in a Swift Playground.

Open Xcode and create a new playground
(File Menu->New->Playground).

From the list of Playground templates, just select "**Blank**"

Step #2: We'll start by declaring a class that we will inherit from later on. Type the following in your own playground.

```
class Mammal {  
  
    func sleep() {  
        print("sleeping")  
    }  
  
    func eat() {  
        print("eating")  
    }  
  
}  
  
let someMammal = Mammal()  
someMammal.eat()  
someMammal.sleep()
```

Step #3: Now let's say we wanted to model a class after a bear. Rather than building it from scratch, let's subclass the Mammal class and add any additional properties and methods we need.

Type out the following below the code from the previous step.

```
class Bear: Mammal {  
  
    var furColor = ""  
  
    func hibernate() {  
        print("hibernating")  
    }  
  
    override func eat() {  
        print("eating fish")  
    }  
  
}  
  
let aBear = Bear()  
aBear.eat()  
aBear.sleep()  
aBear.hibernate()
```

Step #4: Now we want to model an even more specific type of Bear. Type out the following code below the code from the previous step.

```
class PolarBear: Bear {  
    override init() {  
        super.init()  
        furColor = "white"  
    }  
  
    override func eat() {  
        super.eat()  
        print("also eating seals.")  
    }  
}  
  
let mrPolarBear = PolarBear()  
print(mrPolarBear.furColor)  
mrPolarBear.eat()  
mrPolarBear.sleep()  
mrPolarBear.hibernate()
```

Did you notice that we had to override the `init` method? That's because the `Mammal` class has an `init()` method even though we didn't explicitly write it.

All classes have an `init` (short for "initializer") method and this method is automatically called when you create an object of the class.

Also notice that Xcode forces you to call `super.init()` in your overridden `init` method. The reason is because `PolarBear` is a subclass and there may be some important initialization code in the superclass.

Since `PolarBear` inherits all the methods and properties of the `Bear` and `Mammal` classes, calling `super.init()` ensures that those inherited methods and properties are properly initialized and ready to go.

Step #5: Now we want to model another type of Bear. Type out the following code below the code from the previous step.

```
class GrizzlyBear: Bear {  
    override init() {  
        super.init()  
        furColor = "brown"  
    }  
  
    override func eat() {  
        super.eat()  
        print("also eating berries and  
seeds.")  
    }  
  
}  
  
let mrGrizzly = GrizzlyBear()  
print(mrGrizzly.furColor)  
mrGrizzly.eat()  
mrGrizzly.sleep()  
mrGrizzly.hibernate()
```

You're done! If you completed this exercise, celebrate and let me know on Twitter!

<https://twitter.com/CodeWithChris>

To get the solutions, visit this link:

<https://codewithchris.com/code/Module02Lesson03Playground.zip>