

How to Make an App For Beginners

Module 2
Lesson 5
Worksheet



Welcome!

In this worksheet, you'll practice working with Optionals!

Just in case you don't remember how you got here (or if you need a refresher), Lesson 5 can be found here:

https://www.youtube.com/watch?v=jO7Mog_Usk

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: Declare a Person class. Type out the following code in your playground. Xcode will give you an error but ignore it and type it out fully.

```
class Person {  
    var name:String?  
    var age:Int?  
  
    func getDriversLicense() -> String? {  
        if age ≥ 16 && name != nil {  
            return "C9S02J"  
        }  
        else {  
            return nil  
        }  
    }  
}
```

The Xcode error says that we can't compare the optional age to 16. Since the age property is an optional, we have to unwrap it to get the value inside. Add an exclamation mark to unwrap age before comparing it like this:

```
if age! > 16 && name != nil
```

Step #3: Declare another class below the Person class by typing out the following code:

```
class Cop {  
    func checkLicense(licenseNumber:String) -> Bool {  
        if licenseNumber.count < 6 {  
            return false  
        }  
        else {  
            return true  
        }  
    }  
}
```

Step #4: Type out the following code below the Cop class. Xcode will give you an error. What's wrong?

```
let bob = Person()
let licenseNumber = bob.getDriversLicense()
```

The issue is caused by the following line:

```
if age! > 16 && name != nil
```

Bob's age is nil and this if statement is unwrapping Bob's age property, finding nil and then attempting to compare nil with 16. That's an illegal operation so your app crashes.

What we need to do is check if age is nil before trying to compare it with 16. Add the following code BEFORE that age check.

```
if age == nil {
    return nil
}
```

Your new modified getDriversLicense method will look like this:

```
func getDriversLicense() -> String? {  
    if age == nil {  
        return nil  
    }  
  
    if age! > 16 && name != nil {  
        return "C9S02J"  
    }  
    else {  
        return nil  
    }  
}
```

Step #5: Type out the following code at the END of your playground. Xcode will show you an error. What's wrong?

```
let c = Cop()  
c.checkLicense(licenseNumber: licenseNumber!)
```

The issue is caused by the following line:

```
if licenseNumber.count < 6
```

When we called the checkLicense method, we passed in nil. Now we're trying to call the "count" method on nil. That's an illegal operation so your app crashes.

Since checkLicense explicitly says that it expects a String object as a parameter, we shouldn't need to check the parameter for nil. It should be the responsibility of whoever is calling this method to ensure that they're passing valid parameters into this method call. So before calling checkLicense, let's add a nil check like this:

```
let c = Cop()  
if licenseNumber != nil {  
    c.checkLicense(licenseNumber: licenseNumber!)  
}
```

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

<https://twitter.com/CodeWithChris>

Visit the link below for my code for this worksheet:

<https://codewithchris.com/code/Module02Lesson05Playground.zip>