Review: Final

CS112
Max Luttrell, Fall 2016

1. Which of the following functions is called when a user stops on a picker's row:

```
a) func pickerView(pickerView: UIPickerView,
titleForRow row: Int, forComponent component: Int)
-> String?

b) func pickerView(pickerView: UIPickerView,
didSelectRow row: Int, inComponent component: Int)

c) func numberOfComponentsInPickerView(pickerView:
UIPickerView) -> Int

d) func pickerView(pickerView: UIPickerView,
numberOfRowsInComponent component: Int) -> Int
```

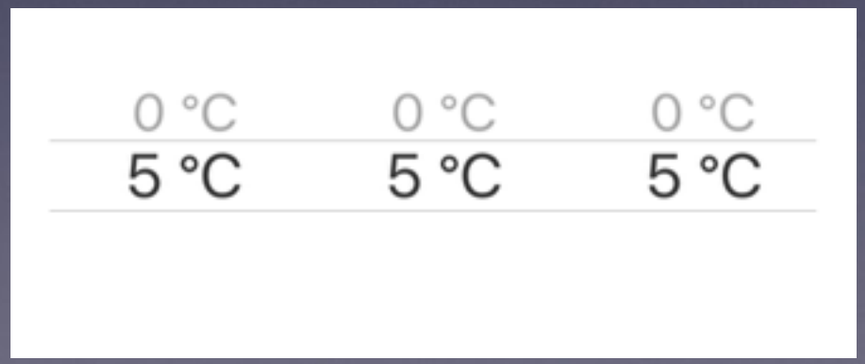
2. How many components does the picker below have?

a) 1 b) 2 c) 3 d) 4

3. How many rows does each component in the picker belowhave?

a) 1 b) 2 c) 3 d) 4

4. True/False: all components in a picker must have the same number of rows



5. A transition between two ViewControllers is called a

a) UIViewControllerTransition

b) viewDidLoad()

c) segue

d) derived ViewController

6. What does the following function call do?

```
dismissViewControllerAnimated(true,
completion: nil)
```

7. True / False. In order to conform to a protocol, a class must be a subclass.

8. True/False In order to conform to a protocol, a class must have a certain minimum set of functions, but can have any properties it wants.

```
class MuniVehicle {
    var totalFares = 0

    func paidFare() {
        totalFares += 2
    }
    func getFares() -> Int {
        return totalFares
    }
}

var c = cableCar()
c.paidFare()          // line 1
print(c.getFares()) // line 2
```

```
class cableCar: MuniVehicle {
    override func getFares() -> Int {
        return totalFares * 3
    }
}
```

9. Consider the above Swift code and choose the best statement below regarding the code commented line 1.

a) this line won't compile because cableCar does not override function paidFare()
b) this line won't compile because cableCar does not have property totalFares
c) this line will increase c's totalFares to 2
d) this line will increase c's totalFares to 6

```
class MuniVehicle {
    var totalFares = 0

    func paidFare() {
        totalFares += 2
    }
    func getFares() -> Int {
        return totalFares
    }
}

var c = cableCar()
c.paidFare()          // line 1
print(c.getFares()) // line 2
```

```
class cableCar: MuniVehicle {
    override func getFares() -> Int {
        return totalFares * 3
    }
}
```

10. Consider the above Swift code and choose the best statement below regarding the code commented line 2.

a) this line won't compile
b) this line will print 2 to the debug area
c) this line will print 6 to the debug area
d) this line will print 0 to the debug area

11. Consider the Swift stub function below which processes an array of Int optionals, representing scores in a daily game. A value of nil indicates that there was no game that day.

Change the function so that it returns the total score, i.e. the sum of elements in the parameter array that are not nil. Then add some code which will call the function with the scores array, and print out the function's return value to the debug area.

```
func totalScore(scores: [Int?]) -> Int {
   return 0
}

let scores: [Int?] = [2, nil, 6, 3, nil, 5, 1, 9, nil, 0]
```

12. Define a Swift class called EvenRange. It should meet the following requirements:

- It should have one property, an empty array of Ints called range.

- There should be a init() method which takes one Int parameter **`numEvens`**. The init() method should not accept values for **`numEvens`** <= 0. If it is called with **`numEvens`** <= 0 an error message should be printed out to the debug area and nothing else should happen. Otherwise, it should add **`numEvens`** even numbers to **`range`**, starting with 0. For example, if **`numEvens`** is 4, init() should change **`range`** to [0, 2, 4, 6]

- It should conform to the **`canPrint`** protocol defined below.

- It should have a function printRange() which will print out the first **`num`** elements in the range to the debug area. If there are fewer than **`num`** elements in the range, it should print the entire range to the debug area. If **`num`** <= 0, it shouldn't print anything. For example, if the range is [0, 2, 4, 6] then printRange(2) should print out 0 2.

```
protocol canPrint {
    func printRange(num: Int)
}
```

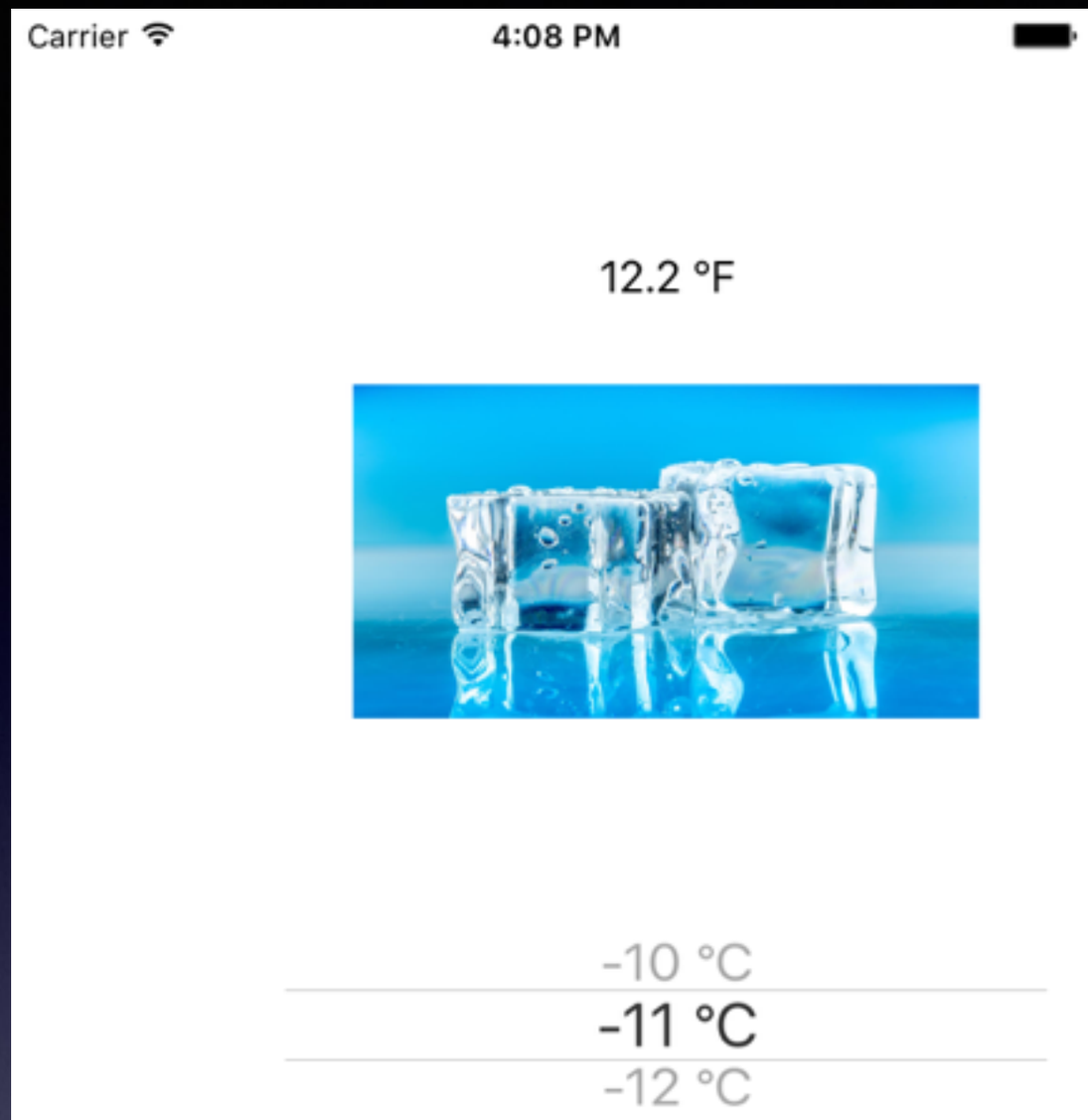
13. Consider the ViewController class on the next page. True/False: this ViewController is from a program which adheres to the MVC design pattern.

a) True b) False

14. Add code as appropriate to display the picker to the right. You may not change any of the code in the next slide, only add code. Requirements:

- picker has 1 component displaying three temps: -10 C, -11 C, -12 C

- when the picker stops on a given row, the label is updated with the equivalent temperature in fahrenheit (no need to change image)



```
class ViewController: UIViewController, UIPickerViewDataSource, UIPickerViewDelegate
{

    @IBOutlet weak var tempLabel: UILabel!
    @IBOutlet weak var imageView: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()

    }

    func numberOfComponentsInPickerView(pickerView: UIPickerView) -> Int {

    }

    func pickerView(pickerView: UIPickerView, numberOfRowsInComponent component:
Int) -> Int {

    }

    func pickerView(pickerView: UIPickerView, titleForRow row: Int, forComponent
component: Int) -> String? {

    }

    func pickerView(pickerView: UIPickerView, didSelectRow row: Int, inComponent
component: Int) {

    }

    func toFahrenheit(celsius: Int) -> String {
        return String(Double(celsius) * 1.8 + 32)
    }
}
```