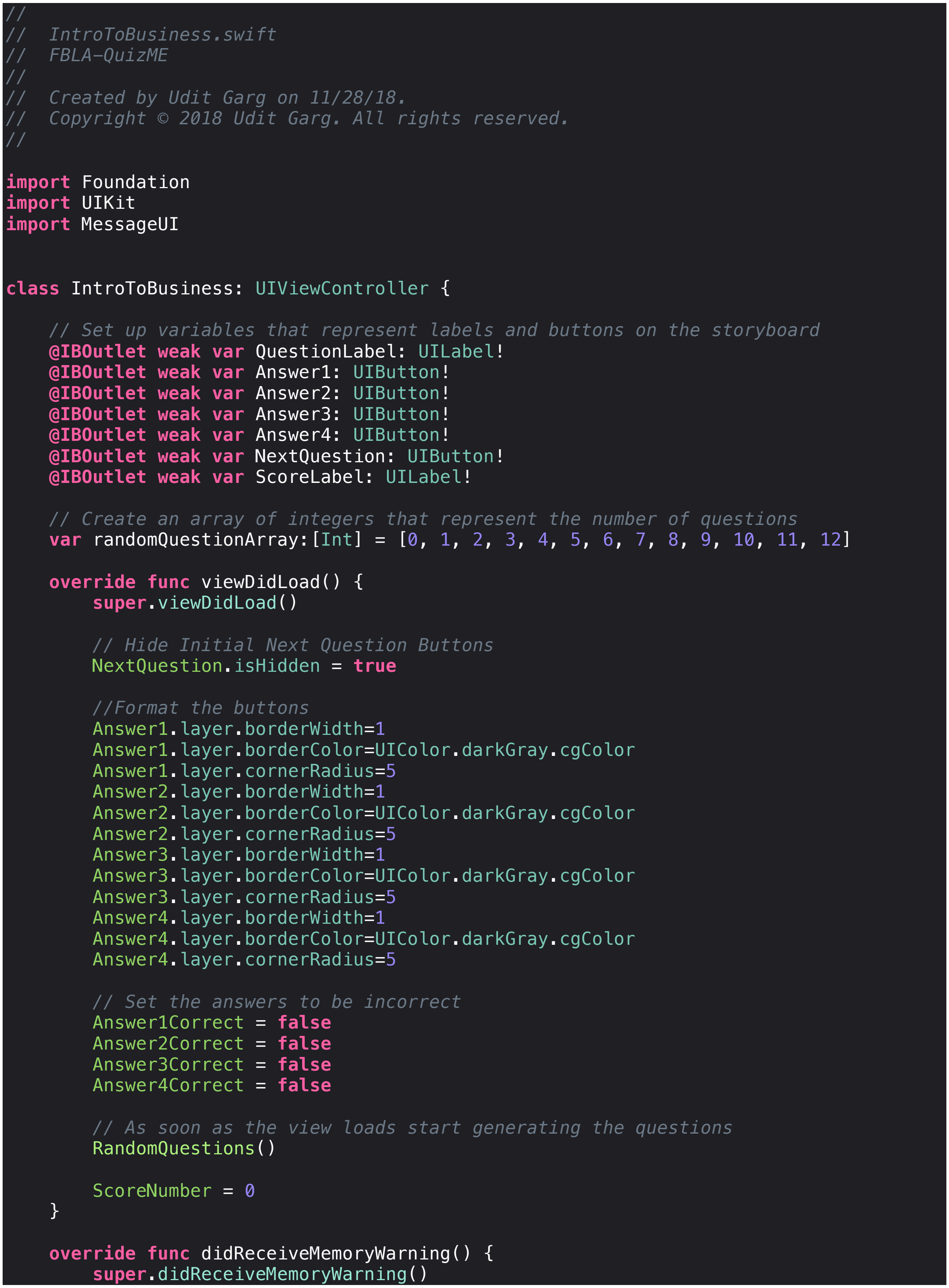
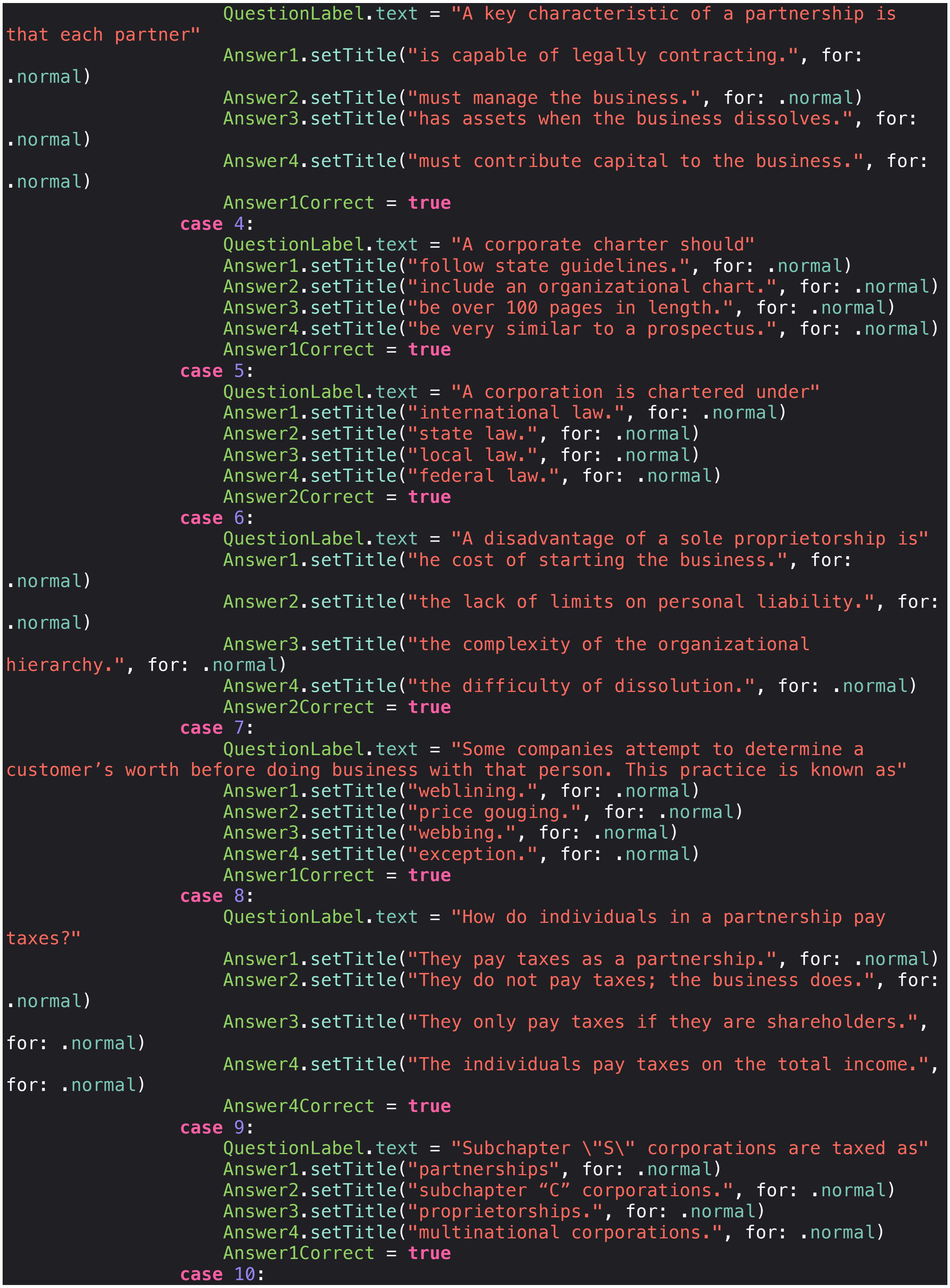
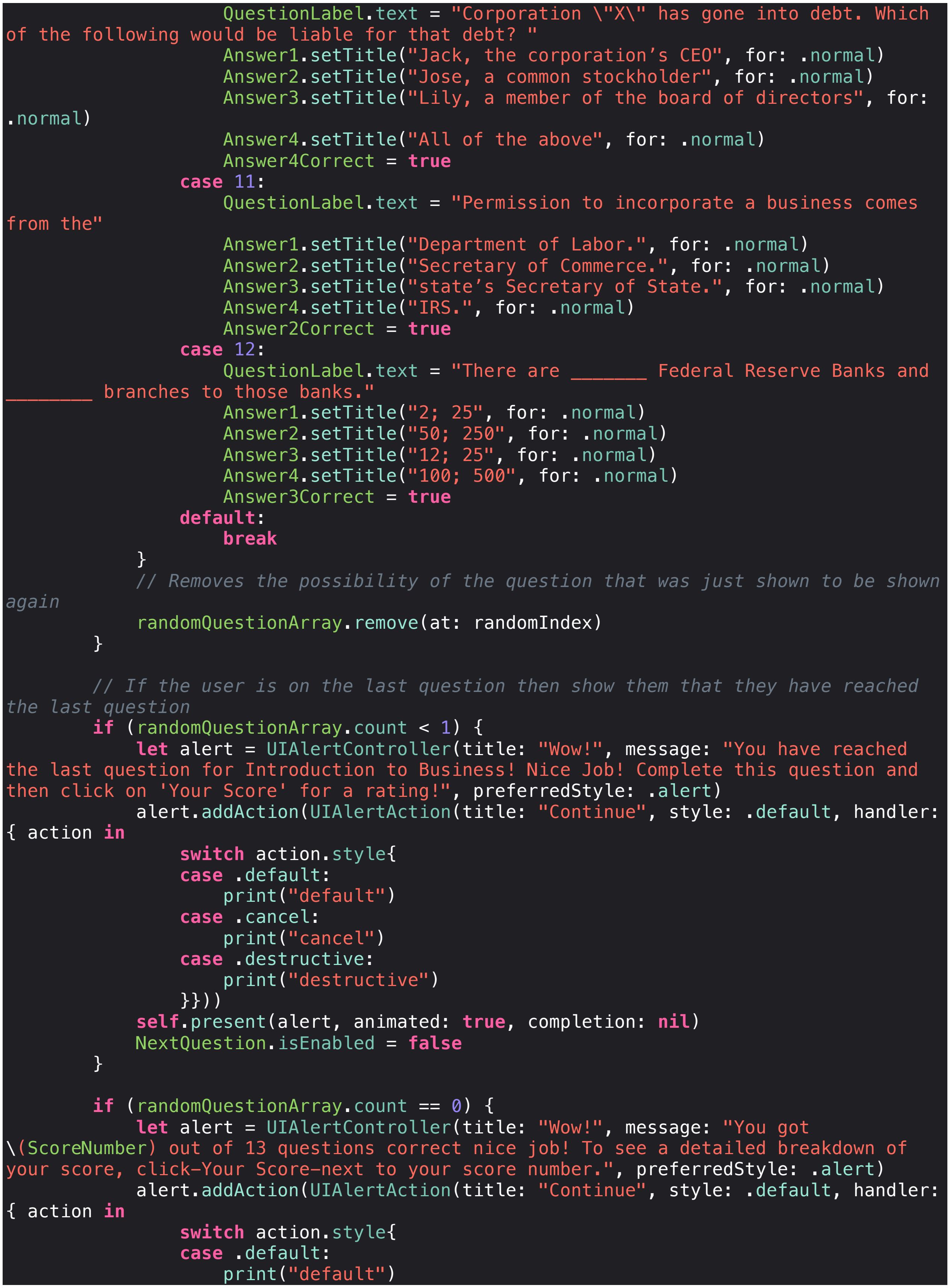


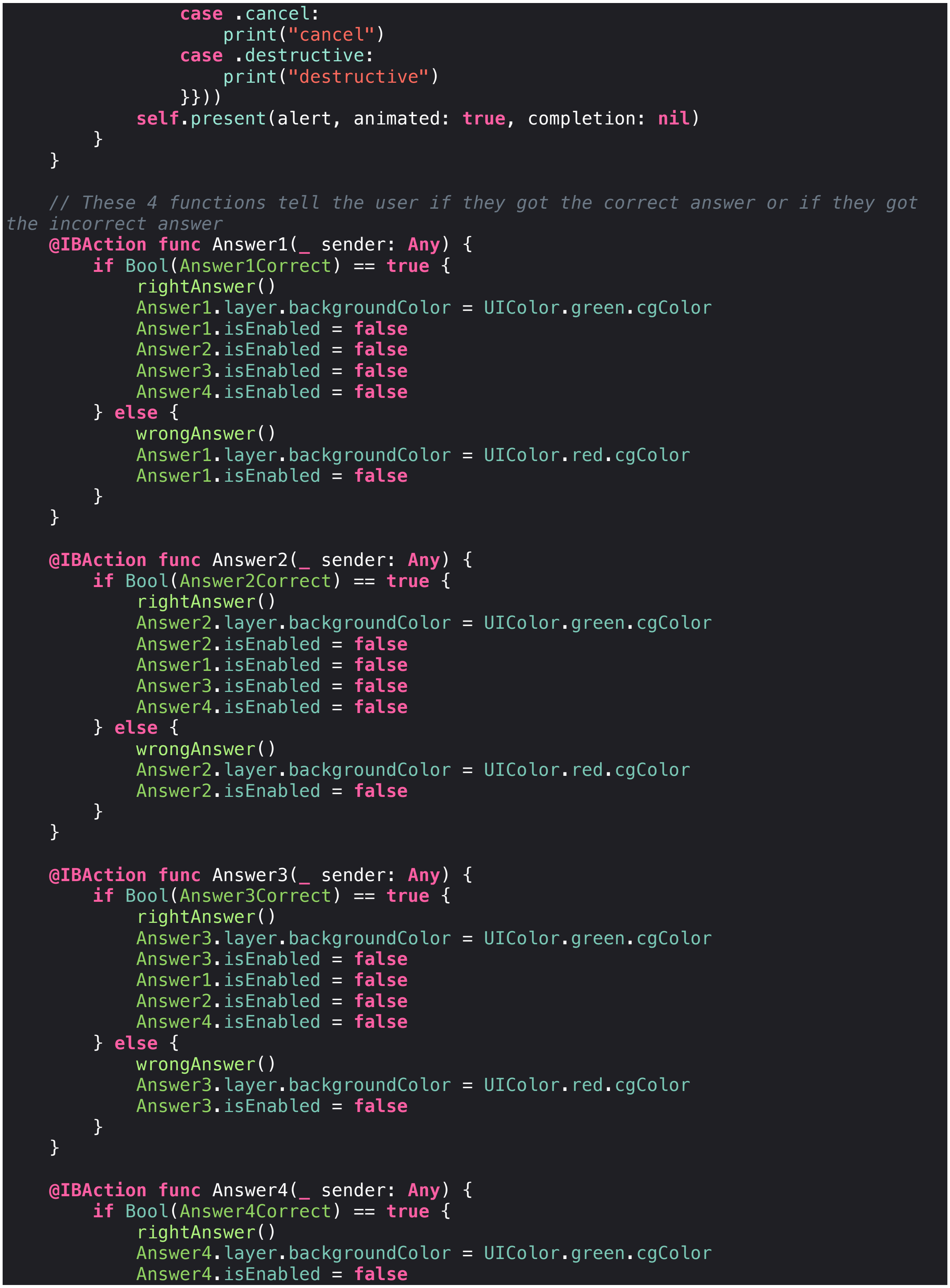
```
//
//  IntroToBusiness.swift
//  FBLA-QuizME
//
//  Created by Udit Garg on 11/28/18.
//  Copyright © 2018 Udit Garg. All rights reserved.
//

import Foundation
import UIKit
import MessageUI


class IntroToBusiness: UIViewController {

    // Set up variables that represent labels and buttons on the storyboard
    @IBOutlet weak var QuestionLabel: UILabel!
    @IBOutlet weak var Answer1: UIButton!
    @IBOutlet weak var Answer2: UIButton!
    @IBOutlet weak var Answer3: UIButton!
    @IBOutlet weak var Answer4: UIButton!
    @IBOutlet weak var NextQuestion: UIButton!
    @IBOutlet weak var ScoreLabel: UILabel!

    // Create an array of integers that represent the number of questions
    var randomQuestionArray:[Int] = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

    override func viewDidLoad() {
        super.viewDidLoad()

        // Hide Initial Next Question Buttons
        NextQuestion.isHidden = true

        //Format the buttons
        Answer1.layer.borderWidth=1
        Answer1.layer.borderColor=UIColor.darkGray.cgColor
        Answer1.layer.cornerRadius=5
        Answer2.layer.borderWidth=1
        Answer2.layer.borderColor=UIColor.darkGray.cgColor
        Answer2.layer.cornerRadius=5
        Answer3.layer.borderWidth=1
        Answer3.layer.borderColor=UIColor.darkGray.cgColor
        Answer3.layer.cornerRadius=5
        Answer4.layer.borderWidth=1
        Answer4.layer.borderColor=UIColor.darkGray.cgColor
        Answer4.layer.cornerRadius=5

        // Set the answers to be incorrect
        Answer1Correct = false
        Answer2Correct = false
        Answer3Correct = false
        Answer4Correct = false

        // As soon as the view loads start generating the questions
        RandomQuestions()

        ScoreNumber = 0
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    // If the answer is right then the Next Question button is enabled
    func rightAnswer() {
        NextQuestion.isHidden = false
        ScoreNumber = Int(ScoreNumber) + 2
        ScoreLabel.text = String(format: "%i", ScoreNumber)
    }

    // If the answer is wrong then the Next Question button is hidden
    func wrongAnswer() {
        ScoreNumber = Int(ScoreNumber) - 1
        ScoreLabel.text = String(format: "%i", ScoreNumber)
    }

    // This function randomly generates questions without repeat
    func RandomQuestions(){

        Answer1.isEnabled = true
        Answer2.isEnabled = true
        Answer3.isEnabled = true
        Answer4.isEnabled = true


        // This makes randomIndex represent the number of questions available for this
question
        let randomIndex = Int(arc4random_uniform(UInt32(randomQuestionArray.count)))

        // Generates questions until all of the questions for this topic have been
answered
        if randomQuestionArray.count > -1 {

            switch (randomQuestionArray[randomIndex]) {
                case 0:
                    QuestionLabel.text = "In a limited partnership, which of the
following remains bound by all debts of the business?"
                    Answer1.setTitle("The partner with the least capital investment",
for: .normal)
                    Answer2.setTitle("Limited partner", for: .normal)
                    Answer3.setTitle("Special partner", for: .normal)
                    Answer4.setTitle("General partner", for: .normal)
                    Answer4Correct = true
                case 1:
                    QuestionLabel.text = "Which of the following would assume all
losses of a business?"
                    Answer1.setTitle("The owner of a sole proprietorship", for:
.normal)
                    Answer2.setTitle("A limited partner", for: .normal)
                    Answer3.setTitle("A stockholder", for: .normal)
                    Answer4.setTitle("The CEO of a corporation", for: .normal)
                    Answer1Correct = true
                case 2:
                    QuestionLabel.text = "Most businesses are"
                    Answer1.setTitle("limited partnerships", for: .normal)
                    Answer2.setTitle("corporations", for: .normal)
                    Answer3.setTitle("general partnerships", for: .normal)
                    Answer4.setTitle("sole proprietorships", for: .normal)
                    Answer4Correct = true
                case 3:
```



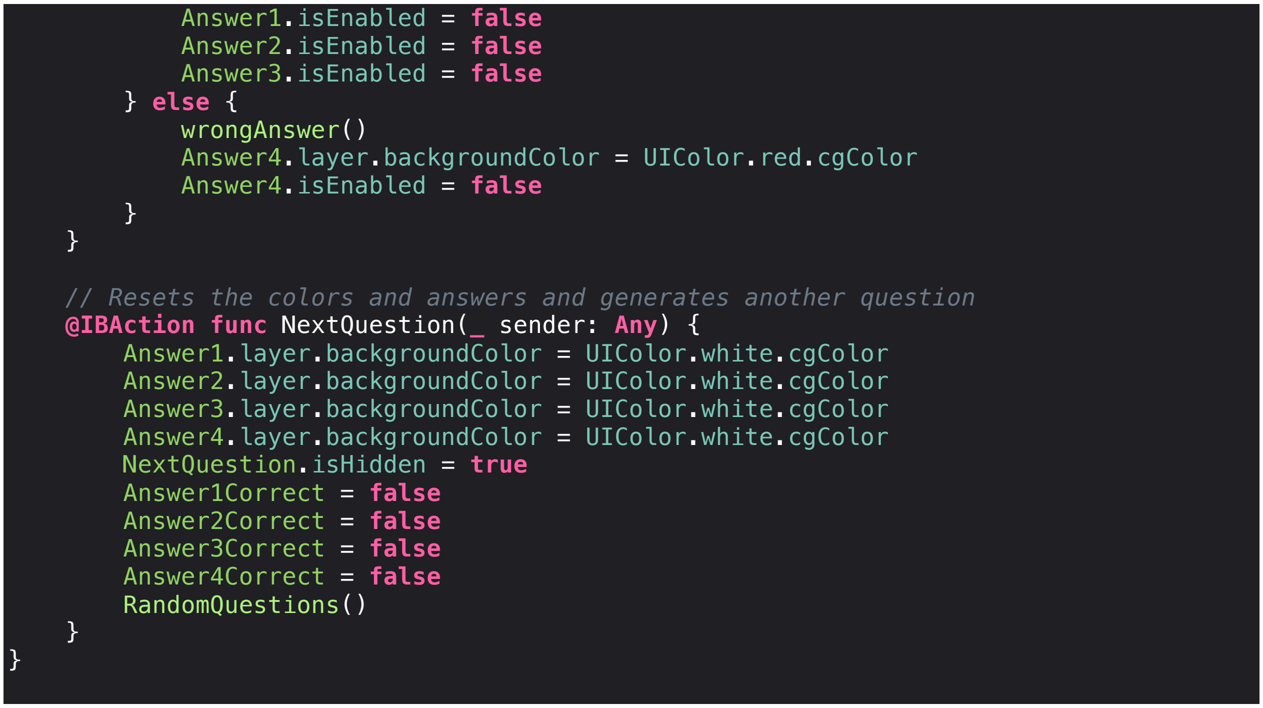
```
                    QuestionLabel.text = "A key characteristic of a partnership is 
that each partner"
                    Answer1.setTitle("is capable of legally contracting.", for: 
.normal)
                    Answer2.setTitle("must manage the business.", for: .normal)
                    Answer3.setTitle("has assets when the business dissolves.", for: 
.normal)
                    Answer4.setTitle("must contribute capital to the business.", for: 
.normal)
                    Answer1Correct = true
                case 4:
                    QuestionLabel.text = "A corporate charter should"
                    Answer1.setTitle("follow state guidelines.", for: .normal)
                    Answer2.setTitle("include an organizational chart.", for: .normal)
                    Answer3.setTitle("be over 100 pages in length.", for: .normal)
                    Answer4.setTitle("be very similar to a prospectus.", for: .normal)
                    Answer1Correct = true
                case 5:
                    QuestionLabel.text = "A corporation is chartered under"
                    Answer1.setTitle("international law.", for: .normal)
                    Answer2.setTitle("state law.", for: .normal)
                    Answer3.setTitle("local law.", for: .normal)
                    Answer4.setTitle("federal law.", for: .normal)
                    Answer2Correct = true
                case 6:
                    QuestionLabel.text = "A disadvantage of a sole proprietorship is"
                    Answer1.setTitle("he cost of starting the business.", for: 
.normal)
                    Answer2.setTitle("the lack of limits on personal liability.", for: 
.normal)
                    Answer3.setTitle("the complexity of the organizational 
hierarchy.", for: .normal)
                    Answer4.setTitle("the difficulty of dissolution.", for: .normal)
                    Answer2Correct = true
                case 7:
                    QuestionLabel.text = "Some companies attempt to determine a 
customer’s worth before doing business with that person. This practice is known as"
                    Answer1.setTitle("weblining.", for: .normal)
                    Answer2.setTitle("price gouging.", for: .normal)
                    Answer3.setTitle("webbing.", for: .normal)
                    Answer4.setTitle("exception.", for: .normal)
                    Answer1Correct = true
                case 8:
                    QuestionLabel.text = "How do individuals in a partnership pay 
taxes?"
                    Answer1.setTitle("They pay taxes as a partnership.", for: .normal)
                    Answer2.setTitle("They do not pay taxes; the business does.", for: 
.normal)
                    Answer3.setTitle("They only pay taxes if they are shareholders.", 
for: .normal)
                    Answer4.setTitle("The individuals pay taxes on the total income.", 
for: .normal)
                    Answer4Correct = true
                case 9:
                    QuestionLabel.text = "Subchapter \"S\" corporations are taxed as"
                    Answer1.setTitle("partnerships", for: .normal)
                    Answer2.setTitle("subchapter “C” corporations.", for: .normal)
                    Answer3.setTitle("proprietorships.", for: .normal)
                    Answer4.setTitle("multinational corporations.", for: .normal)
                    Answer1Correct = true
                case 10:
```



```
                    QuestionLabel.text = "Corporation \"X\" has gone into debt. Which
of the following would be liable for that debt? "
                    Answer1.setTitle("Jack, the corporation’s CEO", for: .normal)
                    Answer2.setTitle("Jose, a common stockholder", for: .normal)
                    Answer3.setTitle("Lily, a member of the board of directors", for:
.normal)
                    Answer4.setTitle("All of the above", for: .normal)
                    Answer4Correct = true
                case 11:
                    QuestionLabel.text = "Permission to incorporate a business comes
from the"
                    Answer1.setTitle("Department of Labor.", for: .normal)
                    Answer2.setTitle("Secretary of Commerce.", for: .normal)
                    Answer3.setTitle("state’s Secretary of State.", for: .normal)
                    Answer4.setTitle("IRS.", for: .normal)
                    Answer2Correct = true
                case 12:
                    QuestionLabel.text = "There are _______ Federal Reserve Banks and
________ branches to those banks."
                    Answer1.setTitle("2; 25", for: .normal)
                    Answer2.setTitle("50; 250", for: .normal)
                    Answer3.setTitle("12; 25", for: .normal)
                    Answer4.setTitle("100; 500", for: .normal)
                    Answer3Correct = true
                default:
                    break
            }
            // Removes the possibility of the question that was just shown to be shown
again
            randomQuestionArray.remove(at: randomIndex)
        }

        // If the user is on the last question then show them that they have reached
the last question
        if (randomQuestionArray.count < 1) {
            let alert = UIAlertController(title: "Wow!", message: "You have reached
the last question for Introduction to Business! Nice Job! Complete this question and
then click on 'Your Score' for a rating!", preferredStyle: .alert)
            alert.addAction(UIAlertAction(title: "Continue", style: .default, handler:
{ action in
                switch action.style{
                case .default:
                    print("default")
                case .cancel:
                    print("cancel")
                case .destructive:
                    print("destructive")
                }}))
            self.present(alert, animated: true, completion: nil)
            NextQuestion.isEnabled = false
        }

        if (randomQuestionArray.count == 0) {
            let alert = UIAlertController(title: "Wow!", message: "You got
\(ScoreNumber) out of 13 questions correct nice job! To see a detailed breakdown of
your score, click-Your Score-next to your score number.", preferredStyle: .alert)
            alert.addAction(UIAlertAction(title: "Continue", style: .default, handler:
{ action in
                switch action.style{
                case .default:
                    print("default")
```



```
                case .cancel:
                    print("cancel")
                case .destructive:
                    print("destructive")
                }}))
            self.present(alert, animated: true, completion: nil)
        }
    }

    // These 4 functions tell the user if they got the correct answer or if they got
the incorrect answer
    @IBAction func Answer1(_ sender: Any) {
        if Bool(Answer1Correct) == true {
            rightAnswer()
            Answer1.layer.backgroundColor = UIColor.green.cgColor
            Answer1.isEnabled = false
            Answer2.isEnabled = false
            Answer3.isEnabled = false
            Answer4.isEnabled = false
        } else {
            wrongAnswer()
            Answer1.layer.backgroundColor = UIColor.red.cgColor
            Answer1.isEnabled = false
        }
    }

    @IBAction func Answer2(_ sender: Any) {
        if Bool(Answer2Correct) == true {
            rightAnswer()
            Answer2.layer.backgroundColor = UIColor.green.cgColor
            Answer2.isEnabled = false
            Answer1.isEnabled = false
            Answer3.isEnabled = false
            Answer4.isEnabled = false
        } else {
            wrongAnswer()
            Answer2.layer.backgroundColor = UIColor.red.cgColor
            Answer2.isEnabled = false
        }
    }

    @IBAction func Answer3(_ sender: Any) {
        if Bool(Answer3Correct) == true {
            rightAnswer()
            Answer3.layer.backgroundColor = UIColor.green.cgColor
            Answer3.isEnabled = false
            Answer1.isEnabled = false
            Answer2.isEnabled = false
            Answer4.isEnabled = false
        } else {
            wrongAnswer()
            Answer3.layer.backgroundColor = UIColor.red.cgColor
            Answer3.isEnabled = false
        }
    }

    @IBAction func Answer4(_ sender: Any) {
        if Bool(Answer4Correct) == true {
            rightAnswer()
            Answer4.layer.backgroundColor = UIColor.green.cgColor
            Answer4.isEnabled = false
```



```
            Answer1.isEnabled = false
            Answer2.isEnabled = false
            Answer3.isEnabled = false
        } else {
            wrongAnswer()
            Answer4.layer.backgroundColor = UIColor.red.cgColor
            Answer4.isEnabled = false
        }
    }

    // Resets the colors and answers and generates another question
    @IBAction func NextQuestion(_ sender: Any) {
        Answer1.layer.backgroundColor = UIColor.white.cgColor
        Answer2.layer.backgroundColor = UIColor.white.cgColor
        Answer3.layer.backgroundColor = UIColor.white.cgColor
        Answer4.layer.backgroundColor = UIColor.white.cgColor
        NextQuestion.isHidden = true
        Answer1Correct = false
        Answer2Correct = false
        Answer3Correct = false
        Answer4Correct = false
        RandomQuestions()
    }
}
```