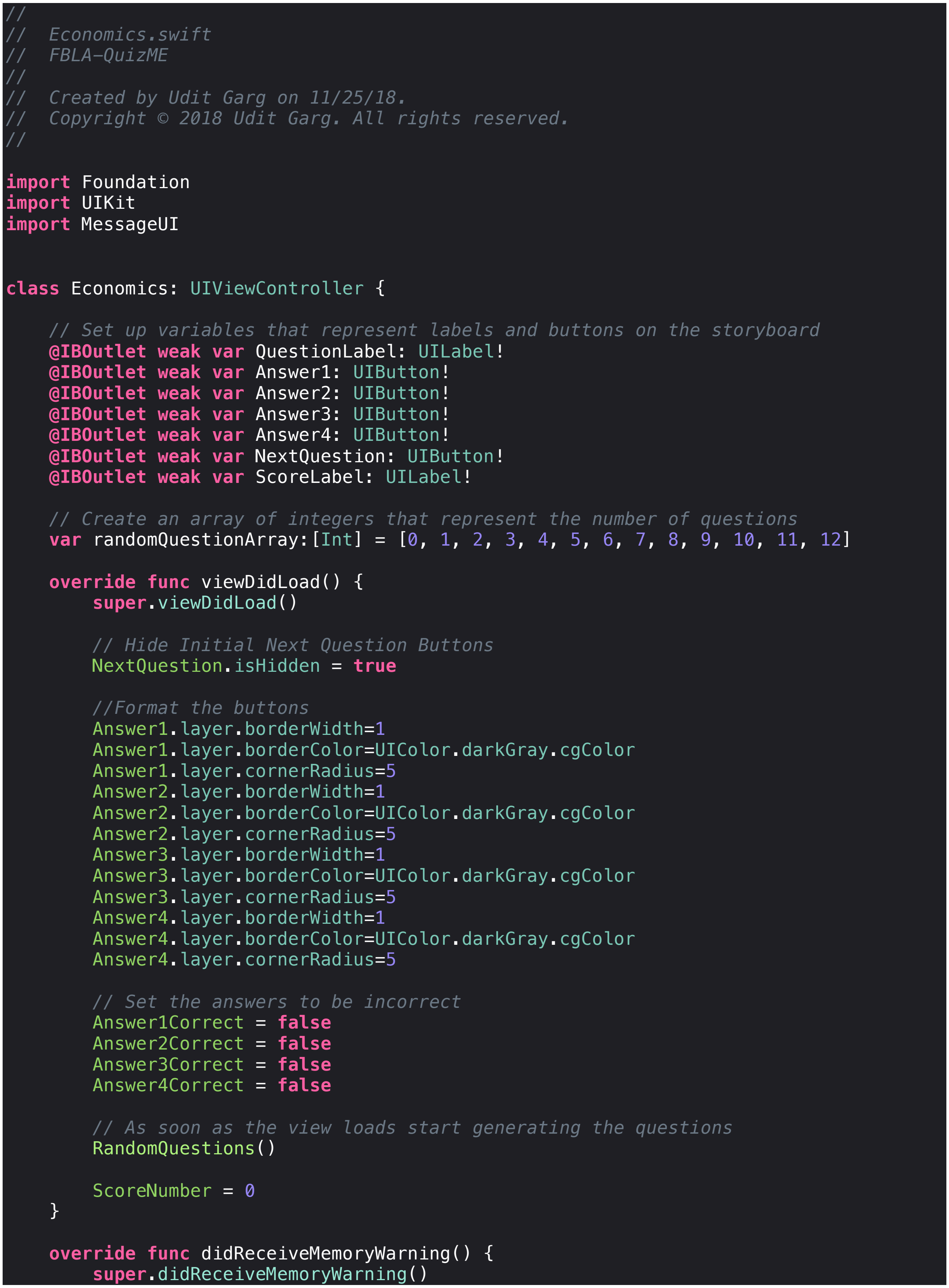
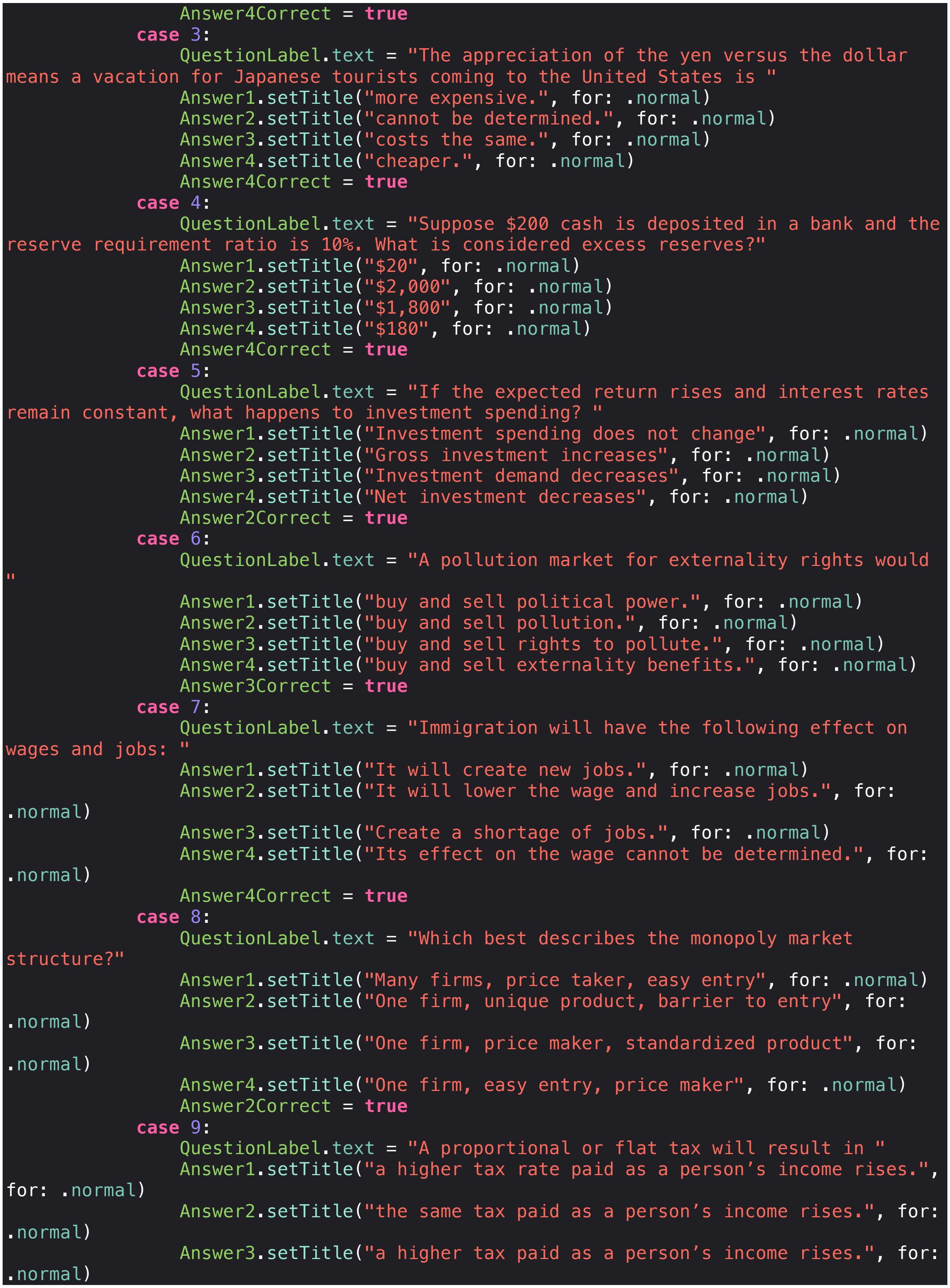
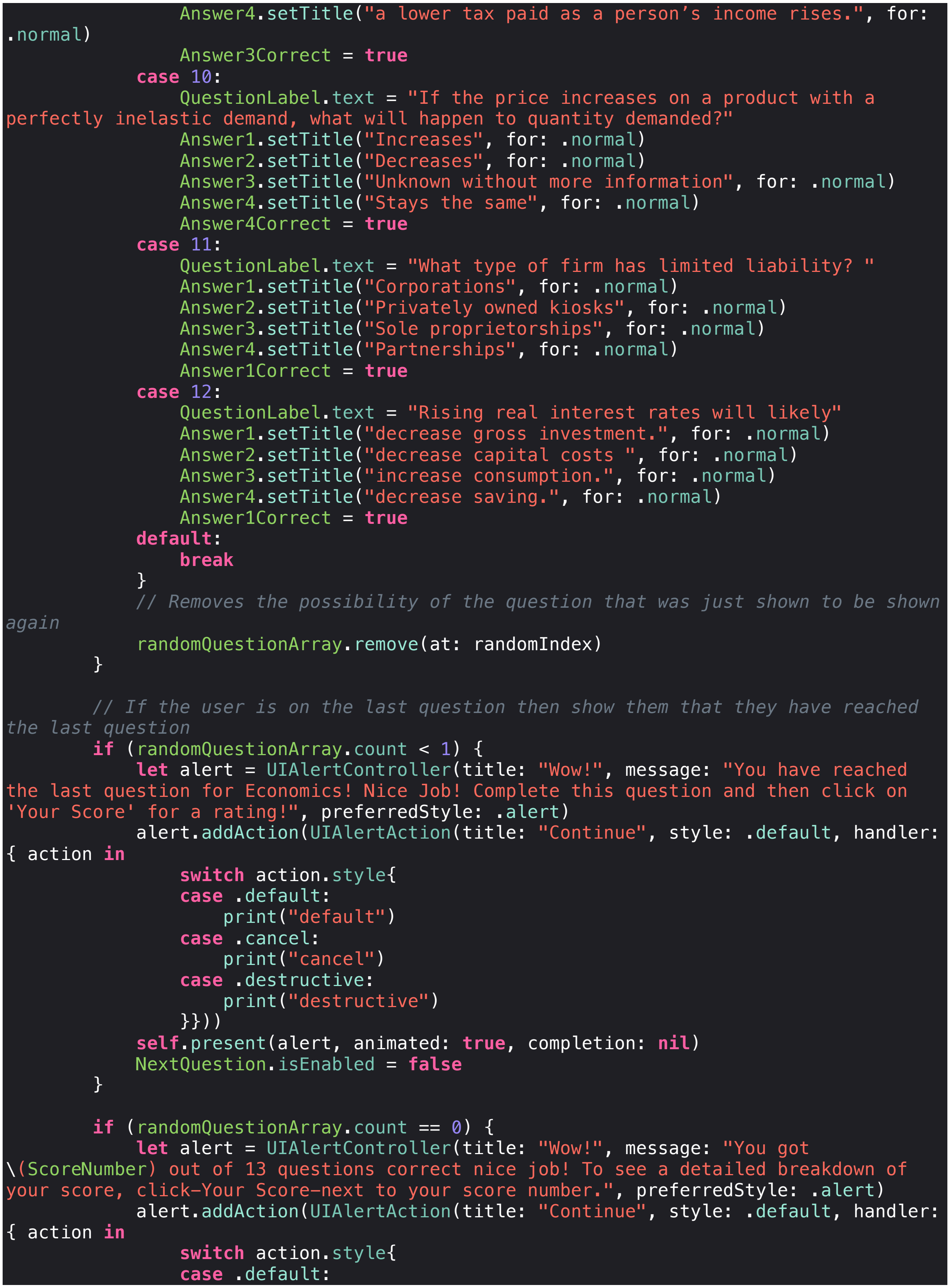


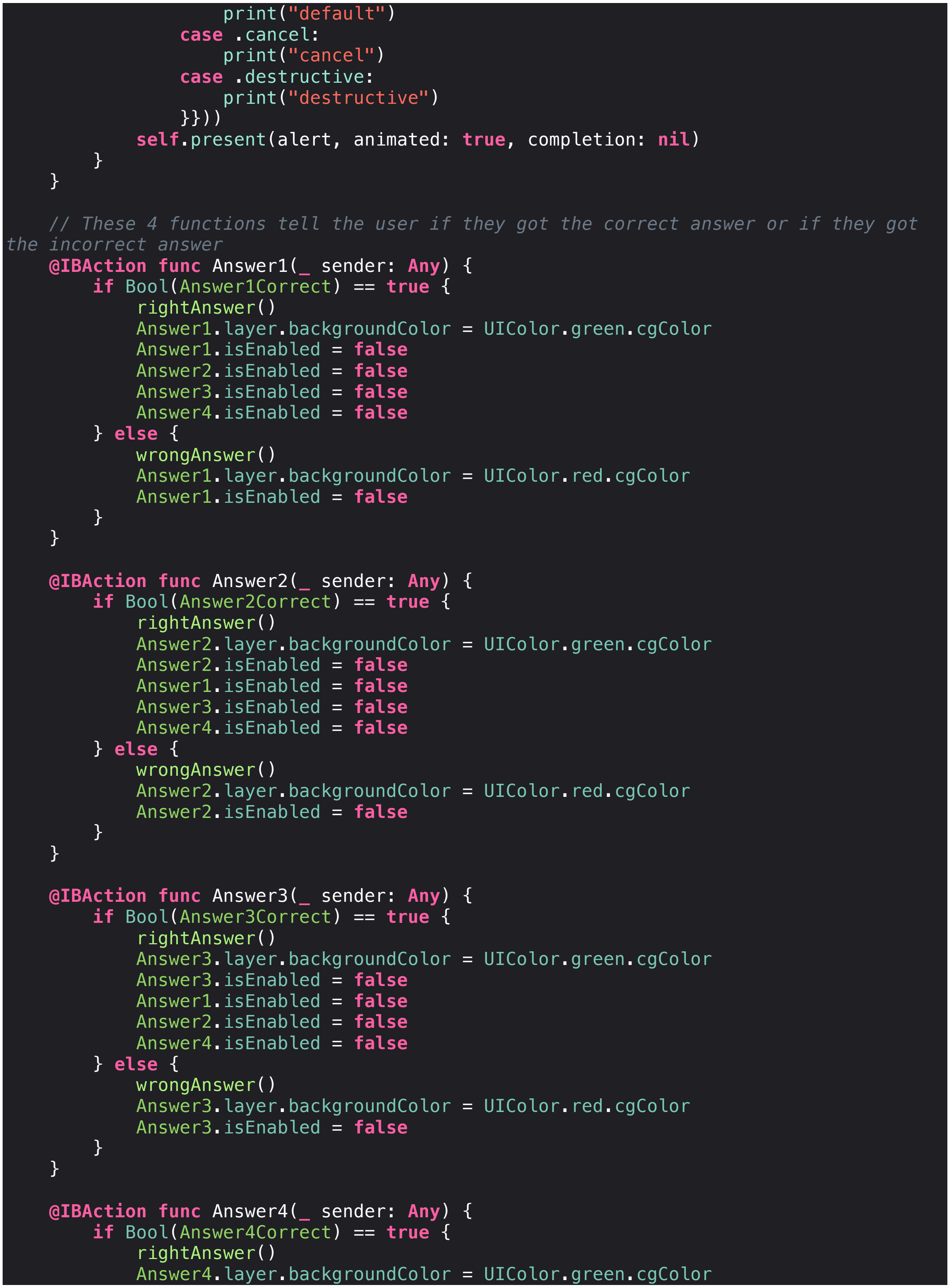
```
//
//  Economics.swift
//  FBLA-QuizME
//
//  Created by Udit Garg on 11/25/18.
//  Copyright © 2018 Udit Garg. All rights reserved.
//

import Foundation
import UIKit
import MessageUI


class Economics: 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 = "Suppose a price floor is placed on wheat, what
is the likely effect? "
                Answer1.setTitle("A lower price", for: .normal)
                Answer2.setTitle("A surplus", for: .normal)
                Answer3.setTitle("Less quantity available", for: .normal)
                Answer4.setTitle("A Shortage", for: .normal)
                Answer2Correct = true
            case 1:
                QuestionLabel.text = "When supply decreases, the equilibrium price and
quantity will"
                Answer1.setTitle("increase in price, decrease in quantity.", for:
.normal)
                Answer2.setTitle("decrease in price, increase in quantity.", for:
.normal)
                Answer3.setTitle("both increase.", for: .normal)
                Answer4.setTitle("both decrease.", for: .normal)
                Answer1Correct = true
            case 2:
                QuestionLabel.text = "Most trade occurs between"
                Answer1.setTitle("poor counties and industrializing countries.", for:
.normal)
                Answer2.setTitle("rich countries and poor countries.", for: .normal)
                Answer3.setTitle("poor counties and other poor countries.", for:
.normal)
                Answer4.setTitle("rich countries and other rich countries.", for:
.normal)
```



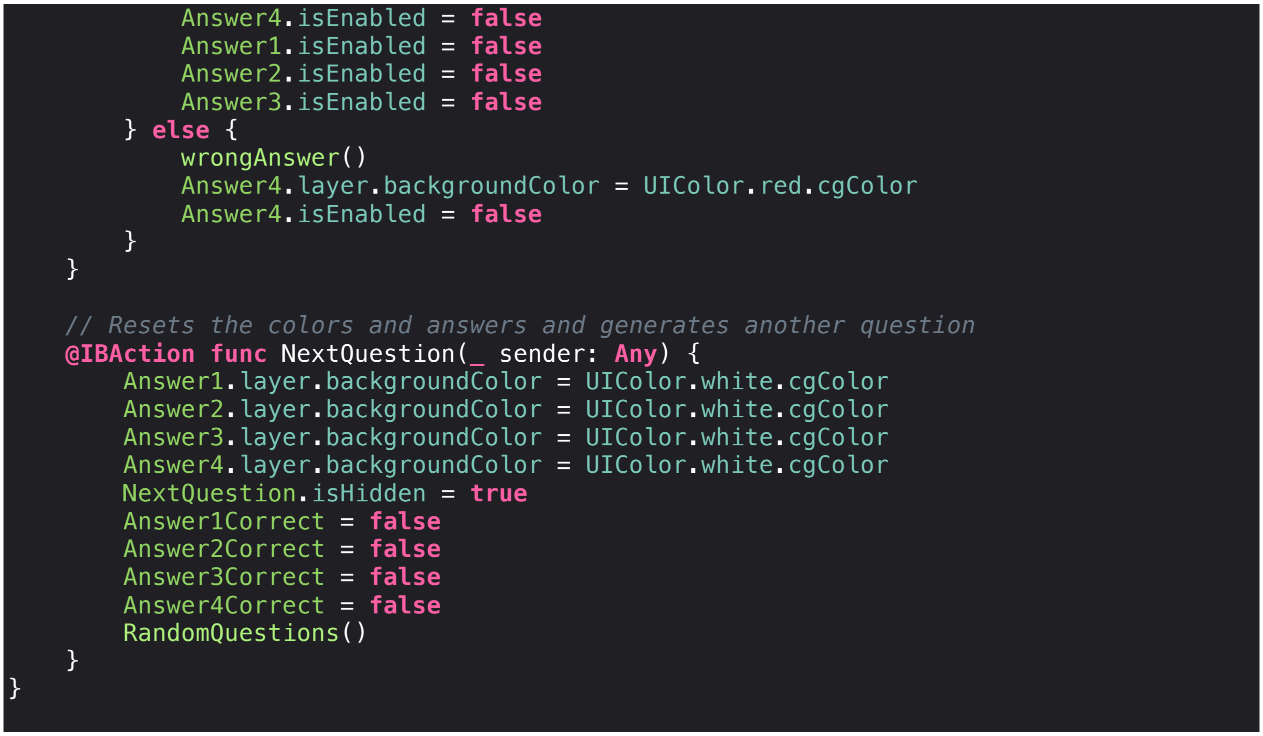
```
                Answer4Correct = true
            case 3:
                QuestionLabel.text = "The appreciation of the yen versus the dollar
means a vacation for Japanese tourists coming to the United States is "
                Answer1.setTitle("more expensive.", for: .normal)
                Answer2.setTitle("cannot be determined.", for: .normal)
                Answer3.setTitle("costs the same.", for: .normal)
                Answer4.setTitle("cheaper.", for: .normal)
                Answer4Correct = true
            case 4:
                QuestionLabel.text = "Suppose $200 cash is deposited in a bank and the
reserve requirement ratio is 10%. What is considered excess reserves?"
                Answer1.setTitle("$20", for: .normal)
                Answer2.setTitle("$2,000", for: .normal)
                Answer3.setTitle("$1,800", for: .normal)
                Answer4.setTitle("$180", for: .normal)
                Answer4Correct = true
            case 5:
                QuestionLabel.text = "If the expected return rises and interest rates
remain constant, what happens to investment spending? "
                Answer1.setTitle("Investment spending does not change", for: .normal)
                Answer2.setTitle("Gross investment increases", for: .normal)
                Answer3.setTitle("Investment demand decreases", for: .normal)
                Answer4.setTitle("Net investment decreases", for: .normal)
                Answer2Correct = true
            case 6:
                QuestionLabel.text = "A pollution market for externality rights would
"
                Answer1.setTitle("buy and sell political power.", for: .normal)
                Answer2.setTitle("buy and sell pollution.", for: .normal)
                Answer3.setTitle("buy and sell rights to pollute.", for: .normal)
                Answer4.setTitle("buy and sell externality benefits.", for: .normal)
                Answer3Correct = true
            case 7:
                QuestionLabel.text = "Immigration will have the following effect on
wages and jobs: "
                Answer1.setTitle("It will create new jobs.", for: .normal)
                Answer2.setTitle("It will lower the wage and increase jobs.", for:
.normal)
                Answer3.setTitle("Create a shortage of jobs.", for: .normal)
                Answer4.setTitle("Its effect on the wage cannot be determined.", for:
.normal)
                Answer4Correct = true
            case 8:
                QuestionLabel.text = "Which best describes the monopoly market
structure?"
                Answer1.setTitle("Many firms, price taker, easy entry", for: .normal)
                Answer2.setTitle("One firm, unique product, barrier to entry", for:
.normal)
                Answer3.setTitle("One firm, price maker, standardized product", for:
.normal)
                Answer4.setTitle("One firm, easy entry, price maker", for: .normal)
                Answer2Correct = true
            case 9:
                QuestionLabel.text = "A proportional or flat tax will result in "
                Answer1.setTitle("a higher tax rate paid as a person’s income rises.",
for: .normal)
                Answer2.setTitle("the same tax paid as a person’s income rises.", for:
.normal)
                Answer3.setTitle("a higher tax paid as a person’s income rises.", for:
.normal)
```



```
        Answer4.setTitle("a lower tax paid as a person’s income rises.", for:
.normal)
        Answer3Correct = true
      case 10:
        QuestionLabel.text = "If the price increases on a product with a
perfectly inelastic demand, what will happen to quantity demanded?"
        Answer1.setTitle("Increases", for: .normal)
        Answer2.setTitle("Decreases", for: .normal)
        Answer3.setTitle("Unknown without more information", for: .normal)
        Answer4.setTitle("Stays the same", for: .normal)
        Answer4Correct = true
      case 11:
        QuestionLabel.text = "What type of firm has limited liability? "
        Answer1.setTitle("Corporations", for: .normal)
        Answer2.setTitle("Privately owned kiosks", for: .normal)
        Answer3.setTitle("Sole proprietorships", for: .normal)
        Answer4.setTitle("Partnerships", for: .normal)
        Answer1Correct = true
      case 12:
        QuestionLabel.text = "Rising real interest rates will likely"
        Answer1.setTitle("decrease gross investment.", for: .normal)
        Answer2.setTitle("decrease capital costs ", for: .normal)
        Answer3.setTitle("increase consumption.", for: .normal)
        Answer4.setTitle("decrease saving.", for: .normal)
        Answer1Correct = 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 Economics! 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()
    }
}
```