```swift
//
//  SecuritiesAndInvestments.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 SecuritiesAndInvestments: 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 = "Brokerage commissions that typically apply to
the small transactions usually made by individual investors are based on"
                Answer1.setTitle("marginal performance", for: .normal)
                Answer2.setTitle("negotiated commissions", for: .normal)
                Answer3.setTitle("fixed-commission schedules", for: .normal)
                Answer4.setTitle("percentage of earnings", for: .normal)
                Answer3Correct = true
            case 1:
                QuestionLabel.text = "The chance that an investment’s value will
decrease is referred to as"
                Answer1.setTitle("load", for: .normal)
                Answer2.setTitle("will", for: .normal)
                Answer3.setTitle("trust", for: .normal)
                Answer4.setTitle("risk", for: .normal)
                Answer4Correct = true
            case 2:
                QuestionLabel.text = "The savings accumulated in a permanent life
insurance policy that you would receive if you canceled your policy is"
                Answer1.setTitle("gross value", for: .normal)
                Answer2.setTitle("cash value", for: .normal)
                Answer3.setTitle("adjusted value", for: .normal)
                Answer4.setTitle("par value", for: .normal)
                Answer2Correct = true
            case 3:
                QuestionLabel.text = "Interest on ___ can be deducted on income taxes"
                Answer1.setTitle("car loans", for: .normal)
```

```
                Answer2.setTitle("revolving credit cards", for: .normal)
                Answer3.setTitle("department store credit card charges", for: .normal)
                Answer4.setTitle("mortgages.", for: .normal)
                Answer4Correct = true
            case 4:
                QuestionLabel.text = "___ security certificates are issued in the
brokerage firm's name but held in trust for its client, who actually owns them."
                Answer1.setTitle("Street name", for: .normal)
                Answer2.setTitle("Day trader", for: .normal)
                Answer3.setTitle("Market order", for: .normal)
                Answer4.setTitle("Insured", for: .normal)
                Answer1Correct = true
            case 5:
                QuestionLabel.text = "Stock dividends stated as a percentage of the
current stock price are referred to as "
                Answer1.setTitle("debenture", for: .normal)
                Answer2.setTitle("market order", for: .normal)
                Answer3.setTitle("percent yield", for: .normal)
                Answer4.setTitle("market value", for: .normal)
                Answer3Correct = true
            case 6:
                QuestionLabel.text = "___ stocks receive dividends first before other
forms of stock"
                Answer1.setTitle("Participating", for: .normal)
                Answer2.setTitle("Blue chip", for: .normal)
                Answer3.setTitle("Common", for: .normal)
                Answer4.setTitle("Preferred", for: .normal)
                Answer4Correct = true
            case 7:
                QuestionLabel.text = "Shares in a company whose earnings are expected
to grow at an above-average rate relative to the market represent a"
                Answer1.setTitle("high risk company", for: .normal)
                Answer2.setTitle("defensive company", for: .normal)
                Answer3.setTitle("growth company", for: .normal)
                Answer4.setTitle("conservative company", for: .normal)
                Answer3Correct = true
            case 8:
                QuestionLabel.text = "The ___ measures inflation/deflation for basic
consumer goods and services"
                Answer1.setTitle("collateralized mortgage operation", for: .normal)
                Answer2.setTitle("Gross Domestic Product", for: .normal)
                Answer3.setTitle("Consumer Product Index", for: .normal)
                Answer4.setTitle("inflation index", for: .normal)
                Answer3Correct = true
            case 9:
                QuestionLabel.text = "A mutual fund that tries to match the
performance of a particular securities index by investing in the companies included in
that index is a(n)"
                Answer1.setTitle("balance fund", for: .normal)
                Answer2.setTitle("index fund", for: .normal)
                Answer3.setTitle("global fund", for: .normal)
                Answer4.setTitle("growth fund", for: .normal)
                Answer2Correct = true
            case 10:
                QuestionLabel.text = "This investment product offers diversity and
professional management."
                Answer1.setTitle("stocks", for: .normal)
                Answer2.setTitle("corporate stock", for: .normal)
                Answer3.setTitle("mutual fund", for: .normal)
                Answer4.setTitle("bonds", for: .normal)
                Answer3Correct = true
```

```
            case 11:
                QuestionLabel.text = "___ can be exchanged for a certain number of
shares of the issuer's common stock."
                Answer1.setTitle("Secured Bonds", for: .normal)
                Answer2.setTitle("Debenture bonds", for: .normal)
                Answer3.setTitle("Savings bonds", for: .normal)
                Answer4.setTitle("Convertible bonds", for: .normal)
                Answer4Correct = true
            case 12:
                QuestionLabel.text = "___ bonds are backed by collateral."
                Answer1.setTitle("Secured", for: .normal)
                Answer2.setTitle("Corporate", for: .normal)
                Answer3.setTitle("Speculative", for: .normal)
                Answer4.setTitle("Unsecured", 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 Securities and Investments! 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)
        }
    }
```

```swift
    // 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()
    }
}
```