```swift
//
//  NetworkingConcepts.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 NetworkingConcepts: 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 = "What is the most common form of copper network
cabling?"
                Answer1.setTitle("Category 5", for: .normal)
                Answer2.setTitle("Category 7", for: .normal)
                Answer3.setTitle("Category 3", for: .normal)
                Answer4.setTitle("Category 4", for: .normal)
                Answer1Correct = true
            case 1:
                QuestionLabel.text = "100BaseFX refers to what cable type? "
                Answer1.setTitle("Wireless connectivity ", for: .normal)
                Answer2.setTitle("Fiber optic cable", for: .normal)
                Answer3.setTitle("Coaxial cable", for: .normal)
                Answer4.setTitle("UTP Cable", for: .normal)
                Answer2Correct = true
            case 2:
                QuestionLabel.text = "What is a valid data rate for a Token Ring
network?"
                Answer1.setTitle("16mbps", for: .normal)
                Answer2.setTitle("100mbps", for: .normal)
                Answer3.setTitle("1000mbps", for: .normal)
                Answer4.setTitle("10mpbs", for: .normal)
                Answer1Correct = true
            case 3:
                QuestionLabel.text = "Which OSI layer is responsible for defining the
format used to exchange data among networked computers?"
                Answer1.setTitle("Session", for: .normal)
```

```
                Answer2.setTitle("Network", for: .normal)
                Answer3.setTitle("Presentation", for: .normal)
                Answer4.setTitle("Physical.", for: .normal)
                Answer3Correct = true
            case 4:
                QuestionLabel.text = "Which of the following fiber devices can act as
a router for routable protocols and also act as a bridge for nonroutable protocols? "
                Answer1.setTitle("Router", for: .normal)
                Answer2.setTitle("Bridge", for: .normal)
                Answer3.setTitle("Switch", for: .normal)
                Answer4.setTitle("Brouter", for: .normal)
                Answer4Correct = true
            case 5:
                QuestionLabel.text = "A MAC address refers to "
                Answer1.setTitle("a series of jumpers on a network card.", for:
.normal)
                Answer2.setTitle("the serial number of a network card.", for: .normal)
                Answer3.setTitle("a unique number assigned to a network device.", for:
.normal)
                Answer4.setTitle("a memory location on a network card.", for: .normal)
                Answer3Correct = true
            case 6:
                QuestionLabel.text = "A CSU/DSU does what function?"
                Answer1.setTitle("Acts as translator between a LAN and a WAN ", for:
.normal)
                Answer2.setTitle("Acts as switching device on a LAN", for: .normal)
                Answer3.setTitle("Terminates a data signal", for: .normal)
                Answer4.setTitle("Routes data", for: .normal)
                Answer1Correct = true
            case 7:
                QuestionLabel.text = "Windows 95, Windows 98, and MacOS9 are all
examples of what?"
                Answer1.setTitle("Peer-to-Peer based operating systems ", for:
.normal)
                Answer2.setTitle("Microsoft operating systems", for: .normal)
                Answer3.setTitle("DOS based operating systems", for: .normal)
                Answer4.setTitle("Server based operating systems", for: .normal)
                Answer1Correct = true
            case 8:
                QuestionLabel.text = "To allow many workstations to access the
Internet through one registered “live” IP address, which service should be used?"
                Answer1.setTitle("Proxy service ", for: .normal)
                Answer2.setTitle("NAT", for: .normal)
                Answer3.setTitle("DNS", for: .normal)
                Answer4.setTitle("One firm, easy entry, price maker", for: .normal)
                Answer2Correct = true
            case 9:
                QuestionLabel.text = "The port number assigned for the POP3 protocol
is "
                Answer1.setTitle("110", for: .normal)
                Answer2.setTitle("43", for: .normal)
                Answer3.setTitle("80", for: .normal)
                Answer4.setTitle("25", for: .normal)
                Answer1Correct = true
            case 10:
                QuestionLabel.text = "Which protocol can be used to transfer a file
between a Unix server and a Windows 2000 server?"
                Answer1.setTitle("FTP", for: .normal)
                Answer2.setTitle("Telnet", for: .normal)
                Answer3.setTitle("PPTP", for: .normal)
                Answer4.setTitle("PPP", for: .normal)
```

```
                Answer1Correct = true
            case 11:
                QuestionLabel.text = "Which of the following networking standards
specifies a maximum segment length of 100 meters?"
                Answer1.setTitle("10BaseX", for: .normal)
                Answer2.setTitle("10Base2", for: .normal)
                Answer3.setTitle("10BaseTX", for: .normal)
                Answer4.setTitle("10Base5", for: .normal)
                Answer3Correct = true
            case 12:
                QuestionLabel.text = "A T3 lines offers transmission speeds up to"
                Answer1.setTitle("6.312Mbps", for: .normal)
                Answer2.setTitle("312.412Mbps", for: .normal)
                Answer3.setTitle("44.736Mbps", for: .normal)
                Answer4.setTitle("274.176Mbps", 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 Networking Concepts! 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()
    }
}
```