```swift
//
//  ComputerProblemSolving.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 ComputerProblemSolving: 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 = "To open the Format cells dialog box, press:"
                Answer1.setTitle("[Ctrl]+[1]", for: .normal)
                Answer2.setTitle("[Ctrl]+[2]", for: .normal)
                Answer3.setTitle("[Ctrl]+[3]", for: .normal)
                Answer4.setTitle("[Ctrl]+[4]", for: .normal)
                Answer1Correct = true
            case 1:
                QuestionLabel.text = "An entry that begins with a letter is recognized
automatically as a:"
                Answer1.setTitle("Formula", for: .normal)
                Answer2.setTitle("Word", for: .normal)
                Answer3.setTitle("Label", for: .normal)
                Answer4.setTitle("Heading", for: .normal)
                Answer3Correct = true
            case 2:
                QuestionLabel.text = "What is the shape of the mouse pointer in the
text area?"
                Answer1.setTitle("An arrow pointing to the left", for: .normal)
                Answer2.setTitle("An arrow pointing to the right", for: .normal)
                Answer3.setTitle("An I-beam", for: .normal)
                Answer4.setTitle("A blinking vertical line", for: .normal)
                Answer3Correct = true
            case 3:
                QuestionLabel.text = "What is the first 640k of memory addresses
called?"
                Answer1.setTitle("extended memory", for: .normal)
```

```
                Answer2.setTitle("upper memory", for: .normal)
                Answer3.setTitle("high memory", for: .normal)
                Answer4.setTitle("conventional memory", for: .normal)
                Answer4Correct = true
            case 4:
                QuestionLabel.text = "Which card is used to add modems and network
cards to the portable computer?"
                Answer1.setTitle("Type 1", for: .normal)
                Answer2.setTitle("Type 2", for: .normal)
                Answer3.setTitle("Type 3", for: .normal)
                Answer4.setTitle("Type 4", for: .normal)
                Answer2Correct = true
            case 5:
                QuestionLabel.text = "Which type of battery is used most often in
notebook computers?"
                Answer1.setTitle("NiMH", for: .normal)
                Answer2.setTitle("NiCad", for: .normal)
                Answer3.setTitle("Li-ION", for: .normal)
                Answer4.setTitle("Zinc Air", for: .normal)
                Answer3Correct = true
            case 6:
                QuestionLabel.text = "Which of the following files is the virtual
memory swap file needed to boot Windows 2000/XP?"
                Answer1.setTitle("Pagefile.sys", for: .normal)
                Answer2.setTitle("Hal.dll", for: .normal)
                Answer3.setTitle("Kernel32.dll", for: .normal)
                Answer4.setTitle("Himem.sys", for: .normal)
                Answer1Correct = true
            case 7:
                QuestionLabel.text = "What is the first cache the CPU uses?"
                Answer1.setTitle("L1 memory", for: .normal)
                Answer2.setTitle("L2 memory", for: .normal)
                Answer3.setTitle("L3 memory", for: .normal)
                Answer4.setTitle("L4 memory", for: .normal)
                Answer1Correct = true
            case 8:
                QuestionLabel.text = "A CMOS virus is an example of this type of
virus"
                Answer1.setTitle("Boot-sector", for: .normal)
                Answer2.setTitle("Trojan", for: .normal)
                Answer3.setTitle("iMule Infector", for: .normal)
                Answer4.setTitle("Malware", for: .normal)
                Answer1Correct = true
            case 9:
                QuestionLabel.text = "What is required when hazardous materials change
hands?"
                Answer1.setTitle("an MSDS", for: .normal)
                Answer2.setTitle("a DSMS", for: .normal)
                Answer3.setTitle("an HDMI", for: .normal)
                Answer4.setTitle("an HTML.", for: .normal)
                Answer1Correct = true
            case 10:
                QuestionLabel.text = "Which version of Win XP only uses the Itanium
processors"
                Answer1.setTitle("Windows XP 8-bit Edition", for: .normal)
                Answer2.setTitle("Windows XP 16-bit Edition", for: .normal)
                Answer3.setTitle("Windows XP 32-bit Edition", for: .normal)
                Answer4.setTitle("Windows XP 64-bit Edition", for: .normal)
                Answer4Correct = true
            case 11:
```

```
                QuestionLabel.text = "The detailed view of a stock offering that must
be filed with the SEC is known as a"
                Answer1.setTitle("prospectusc", for: .normal)
                Answer2.setTitle("registration statement", for: .normal)
                Answer3.setTitle("financial statement", for: .normal)
                Answer4.setTitle("commerce clause", for: .normal)
                Answer1Correct = true
            case 12:
                QuestionLabel.text = "Windows Advanced Server can support up to"
                Answer1.setTitle("8 symmetrical processors and 8M of memory", for:
.normal)
                Answer2.setTitle("4 symmetrical processors and 8M of memory", for:
.normal)
                Answer3.setTitle("8 asymmetrical processors and 8GB of memory", for:
.normal)
                Answer4.setTitle("4 asymmetrical processors and 4GB of memory", 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 Computer Problem Solving! 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()
    }
}
```