

```

//
//  HealthCareAdministration.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 HealthCareAdministration: 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 = "Which one of the following represents a behavior
that is intended to protect oneself from a perceived threat?"
            Answer1.setTitle("defensive", for: .normal)
            Answer2.setTitle("offensive", for: .normal)
            Answer3.setTitle("ethnic", for: .normal)
            Answer4.setTitle("anxiety", for: .normal)
            Answer1Correct = true
        case 1:
            QuestionLabel.text = "Confidentiality is a major concern when"
            Answer1.setTitle("faxing documents to a patient", for: .normal)
            Answer2.setTitle("e-mailing documents to a patient", for: .normal)
            Answer3.setTitle("leaving a message on a patient's answering machine",
for: .normal)
            Answer4.setTitle("talking to other patients about the patient in
cocern", for: .normal)
            Answer3Correct = true
        case 2:
            QuestionLabel.text = "A bruise also is known as a/an"
            Answer1.setTitle("cardio version", for: .normal)
            Answer2.setTitle("contusion", for: .normal)
            Answer3.setTitle("occlusion", for: .normal)
            Answer4.setTitle("cauterization", for: .normal)
            Answer2Correct = true
        case 3:
            QuestionLabel.text = "Which definition best describes a thoracotomy?"
            Answer1.setTitle("incision through the chest wall", for: .normal)

```

```

        Answer2.setTitle("pertaining to the location of the thorax", for:
.normal)
        Answer3.setTitle("delineation of deep structures behind the chest
wall", for: .normal)
        Answer4.setTitle("examination of the chest wall", for: .normal)
        Answer1Correct = true
    case 4:
        QuestionLabel.text = "Patient education is critical to prevent the
onset of a repeat of illness. What is essential to assess before beginning patient
education?"
        Answer1.setTitle("time available for teaching", for: .normal)
        Answer2.setTitle("caregiver's expectations", for: .normal)
        Answer3.setTitle("readiness to learn", for: .normal)
        Answer4.setTitle("ability to integrate medication with new
strategies", for: .normal)
        Answer3Correct = true
    case 5:
        QuestionLabel.text = "The Centers for Disease Control and Prevention
is the principal agency concerned with the health and safety of people throughout the"
        Answer1.setTitle("world", for: .normal)
        Answer2.setTitle("United States", for: .normal)
        Answer3.setTitle("Southwest region", for: .normal)
        Answer4.setTitle("Northwest region", for: .normal)
        Answer1Correct = true
    case 6:
        QuestionLabel.text = "What is the name of a service that may be
employed by a medical practice to prepare and send all monthly statements and to place
collection calls when needed?"
        Answer1.setTitle("billing service", for: .normal)
        Answer2.setTitle("collection agency", for: .normal)
        Answer3.setTitle("service agency", for: .normal)
        Answer4.setTitle("clearinghouse", for: .normal)
        Answer4Correct = true
    case 7:
        QuestionLabel.text = "The government requires businesses to report
incomes on a regular basis. What is the required form to be used in this reporting
process?"
        Answer1.setTitle("wage and tax statement form", for: .normal)
        Answer2.setTitle("wage and taxable account form", for: .normal)
        Answer3.setTitle("employee withholding form", for: .normal)
        Answer4.setTitle("employer quarterly tax reporting form", for:
.normal)
        Answer2Correct = true
    case 8:
        QuestionLabel.text = "Which one of the following statements regarding
interviewing a patient is correct?"
        Answer1.setTitle("Make sure the patient's information is correct",
for: .normal)
        Answer2.setTitle("Let the patient know there is limited time for
interviewing", for: .normal)
        Answer3.setTitle("Interrupt the patient so you have the right
information.", for: .normal)
        Answer4.setTitle("Ask yes/no questions to be sure information is
correct.", for: .normal)
        Answer1Correct = true
    case 9:
        QuestionLabel.text = "Moving your fingers across the screen, instead
of using a mouse is known as which one of the following technologies?"
        Answer1.setTitle("sensitive screen", for: .normal)
        Answer2.setTitle("flash screen", for: .normal)
        Answer3.setTitle("mobile screen", for: .normal)

```

```

        Answer4.setTitle("touch screen", for: .normal)
        Answer4Correct = true
    case 10:
        QuestionLabel.text = "What did the National Labor Relations Act do?"
        Answer1.setTitle("Allowed health care providers to use bargaining
units", for: .normal)
        Answer2.setTitle("Established minimum wage, ages, and overtime pay for
workers", for: .normal)
        Answer3.setTitle("Established pay equality between males and females",
for: .normal)
        Answer4.setTitle("Established guidelines for workplace safety", for:
.normal)
        Answer1Correct = true
    case 11:
        QuestionLabel.text = "What did the Equal Pay Act do?"
        Answer1.setTitle("Allowed health care providers to use bargaining
units", for: .normal)
        Answer2.setTitle("Established minimum wage, ages, and overtime pay for
workers", for: .normal)
        Answer3.setTitle("Established pay equality between males and females",
for: .normal)
        Answer4.setTitle("Established guidelines for workplace safety", for:
.normal)
        Answer3Correct = true
    case 12:
        QuestionLabel.text = "What did the Fair Labor Standards Act of 1938
do?"
        Answer1.setTitle("Allowed health care providers to use bargaining
units", for: .normal)
        Answer2.setTitle("Established minimum wage, ages, and overtime pay for
workers", for: .normal)
        Answer3.setTitle("Established pay equality between males and females",
for: .normal)
        Answer4.setTitle("Established guidelines for workplace safety", for:
.normal)
        Answer2Correct = 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 Health Care Administration! 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()
}
}

```