

```

//
//  Accounting 1.swift
//  FBLA-QuizME
//
//  Created by Udit Garg on 11/22/18.
//  Copyright © 2018 Udit Garg. All rights reserved.
//

import Foundation
import UIKit

// Declare universal variables
var Answer1Correct = false
var Answer2Correct = false
var Answer3Correct = false
var Answer4Correct = false
var ScoreNumber: Int = 0

class Accounting1: 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 fun didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    // If the answer is right then the Next Question button is enabled
    fun 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
    fun wrongAnswer() {
        ScoreNumber = Int(ScoreNumber) - 1
        ScoreLabel.text = String(format: "%i", ScoreNumber)
    }

    // This function randomly generates questions without repeat
    fun 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 = "When cash is received from sales, ____."
                    Answer1.setTitle("assets increase; owner's equity decreases", for:
.normal)
                    Answer2.setTitle("assets increase; owner's equity increases", for:
.normal)
                    Answer3.setTitle("assets decrease; owner's equity decreases", for:
.normal)
                    Answer4.setTitle("none of the above", for: .normal)
                    Answer2Correct = true
                case 1:
                    QuestionLabel.text = "Words in accounting are ____"
                    Answer1.setTitle("written in full when space permits", for:
.normal)
                    Answer2.setTitle("abbreviated wherever possible", for: .normal)
                    Answer3.setTitle("printed rather than written", for: .normal)
                    Answer4.setTitle("none of the above", for: .normal)
                    Answer1Correct = true
                case 2:
                    QuestionLabel.text = "An endorsement on the back of a check
consisting of the words- 'Pay to the order' of and a new check owner's name is a
____."

```

```

        Answer1.setTitle("blank endorsement", for: .normal)
        Answer2.setTitle("special endorsement", for: .normal)
        Answer3.setTitle("restrictive endorsement", for: .normal)
        Answer4.setTitle("signature endorsement", for: .normal)
        Answer2Correct = true
    case 3:
        QuestionLabel.text = "The date on a monthly income statement
prepared on July 31 is written as ____."
        Answer1.setTitle("For Month Ended July 31, 20-", for: .normal)
        Answer2.setTitle("July 31, 20-", for: .normal)
        Answer3.setTitle("20-, July 31", for: .normal)
        Answer4.setTitle("none of the above", for: .normal)
        Answer1Correct = true
    case 4:
        QuestionLabel.text = "A ledger that is summarized in a single
general ledger account is a ____."
        Answer1.setTitle("ledger", for: .normal)
        Answer2.setTitle("controlling ledger", for: .normal)
        Answer3.setTitle("secondary ledger", for: .normal)
        Answer4.setTitle("subsidiary ledger", for: .normal)
        Answer4Correct = true
    case 5:
        QuestionLabel.text = "The total pay due for a pay period before
deductions is ____."
        Answer1.setTitle("gross pay", for: .normal)
        Answer2.setTitle("gross earnings", for: .normal)
        Answer3.setTitle("total earnings", for: .normal)
        Answer4.setTitle("all of the above", for: .normal)
        Answer2Correct = true
    case 6:
        QuestionLabel.text = "Total shares of ownership in a corporation
is ____."
        Answer1.setTitle("a share of stock", for: .normal)
        Answer2.setTitle("capital stock", for: .normal)
        Answer3.setTitle("a trade discount", for: .normal)
        Answer4.setTitle("none of the above", for: .normal)
        Answer2Correct = true
    case 7:
        QuestionLabel.text = "Using the price of merchandise purchased
first to calculate the cost of merchandise sold first is the ____."
        Answer1.setTitle("fifo method", for: .normal)
        Answer2.setTitle("lifo method", for: .normal)
        Answer3.setTitle("gross profit method", for: .normal)
        Answer4.setTitle("weighted-average method", for: .normal)
        Answer1Correct = true
    case 8:
        QuestionLabel.text = "The amount of capital stock issued at the
beginning of the year is obtained from ____."
        Answer1.setTitle("a Balance Sheet Debit column of a work sheet",
for: .normal)
        Answer2.setTitle("a Balance Sheet Credit column of a work sheet",
for: .normal)
        Answer3.setTitle("a general ledger account for capital stock",
for: .normal)
        Answer4.setTitle("none of the above", for: .normal)
        Answer3Correct = true
    case 9:
        QuestionLabel.text = "The person or business to whom the signer of
a promissory note promises a future payment is called the ____."
        Answer1.setTitle("maker", for: .normal)
        Answer2.setTitle("principal party", for: .normal)

```

```

        Answer3.setTitle("payee", for: .normal)
        Answer4.setTitle("drawer", for: .normal)
        Answer3Correct = true
    case 10:
        QuestionLabel.text = "Williams Company accepted a nine-month,
$5,000, 6% note, from Pace Corporation on September 1, 1999. The amount of interest
to be accrued on December 31, 1999 is ____."
        Answer1.setTitle("$100", for: .normal)
        Answer2.setTitle("$225", for: .normal)
        Answer3.setTitle("$300", for: .normal)
        Answer4.setTitle("$75", for: .normal)
        Answer1Correct = true
    case 11:
        QuestionLabel.text = "The entry to journalize the replenishment of
petty cash includes a ____."
        Answer1.setTitle("debit to petty cash", for: .normal)
        Answer2.setTitle("credit to cash", for: .normal)
        Answer3.setTitle("debit to cash", for: .normal)
        Answer4.setTitle("credit to petty cash", for: .normal)
        Answer4Correct = true
    case 12:
        QuestionLabel.text = "Depreciable cost is defined as ____."
        Answer1.setTitle("cost minus accumulated depreciation", for:
.normal)
        Answer2.setTitle("book value", for: .normal)
        Answer3.setTitle("residual value", for: .normal)
        Answer4.setTitle("cost minus scrap value", for: .normal)
        Answer3Correct = 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 Accounting 1! 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()
    }
}

```