6COSC021W Lab Based Practical – Specification – Mock Test

Week 7, Semester 2, 2023

You have <u>1 hours and 45 minutes</u> to complete this assessment. You will be told when you have; 1 hour, 30 minutes, and 15 minutes remaining. Work will be reviewed in the next seminar.

Before the assessment starts:

1) Download the assessment project template from Blackboard.
2) Run the project and ensure there are no errors or build issues.
3) Make sure you have to hand any allowed notes you require. See Appendix B.
4) Spend some time inspecting the starter project. The main structure of the code will be stubbed out with comments that indicate what code needs to be placed in these sections.
5) Familiarise yourself with this brief and ask any question if you are unclear on any point before you begin coding.

Instructions

You are to build one view for Maths app that solves a quadratic equation in SwiftUI.

The user will enter these terms for the standard equation as shown below:

$$Ax^2 + Bx + C = 0$$

The application allows the users to enter:

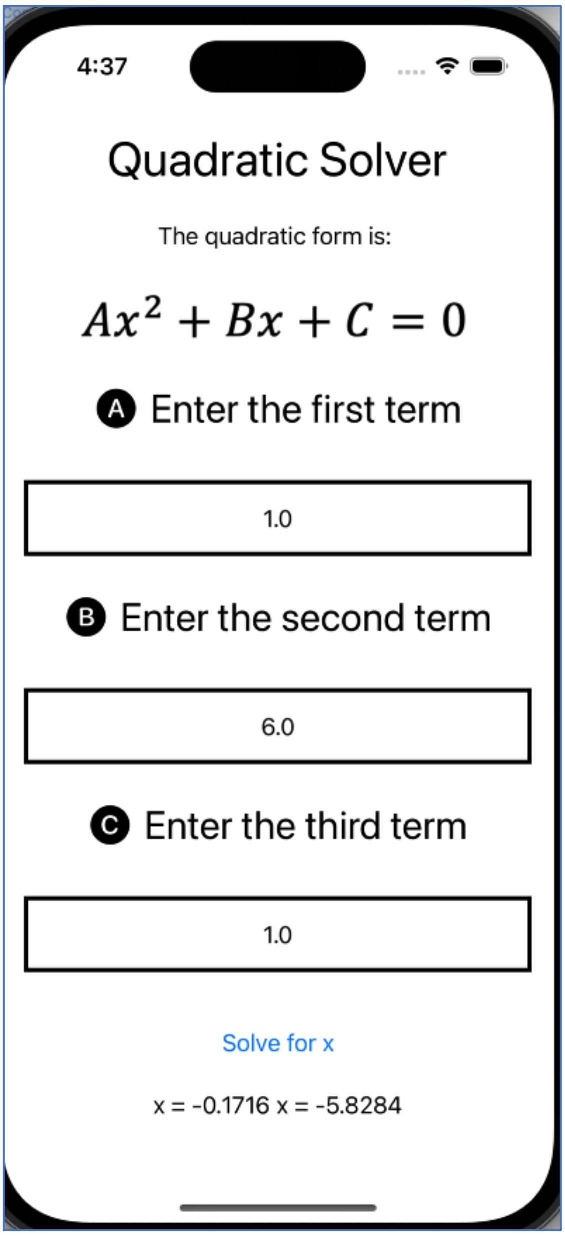
1. ***A*** the quadratic term.
2. ***B*** the linear term.
3. ***C*** the constant term.

The application outputs:

a) The two roots: x1 and x2. See Appendix A.

User Interface

The user interface should look as close as possible to the that shown in figure 1 below.



4:37

Quadratic Solver

The quadratic form is:

$$Ax^2 + Bx + C = 0$$

Ⓐ Enter the first term

1.0

Ⓑ Enter the second term

6.0

Ⓒ Enter the third term

1.0

x = -0.1716 x = -5.8284

Figure 1 User interface for the maths app

Tasks – read all tasks before starting to code

1) Implement and style the UI – *see Figure 1 User interface for the maths app*. Note: do not yet implement the equation image. See Appendix B for SF symbol names for A, B, and C on the labels.
2) Implement the function *solveForX()* that calculates the two roots of the equation. Note use type **double** for all numbers – see Appendix A for the formula.
3) Attached the function to the button so the calculation is performed when the button is touched.
   Once implemented you can test you application with the default data.
   A = 1.0 B = 6.0 and C = 1.0 gives:
   x1 = -0.1716 and x2 = -5.8284

4) Implement the code to ensure that x1 and x2 output strings are formatted to two decimal places.
5) Implement the necessary code to ensure that the any data entered into the text fields is persisted is the application is closed/backgrounded – (You can use @AppStorage)
6) Add the formula image*.*
7) Zip your completed project and upload to Blackboard – **Mock Lab Practical Test** – name your project wXXXXXXX-7SENG-MockPractical. Where wXXXXXXX is your student ID.

Appendix A

Mortgage calculation for monthly payment p is shown in equation (3):

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Note that if $b^2 - 4ac < 0$ the roots will be complex. You do not need to implement or test for this.

Appendix B

Allowed noted are:

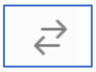
1) Lecture notes (Blackboard site allowed)
2) Seminar notes (Blackboard site allowed)
3) Apple API documentation from within XCode only

Assets

The **SF Symbols** are: **a.circle.fill**, **b.circle.fill,** and **c.circle.fill**.

The equation image is called **quadraticForm** and is stored in the Assets folder and so can be called directly by name.

Note that you cannot see the Assets if you have **Enable Code Review** clicked, so this switch needs to look like the icon below and should not have a blue background.



No code can be obtained from any other source as this will constitute an academic assessment misconduct.

Appendix C

Starter Code - for information only

```
//
//  ContentView.swift
//  QuadraticSolver
//
//  Created by Philip Trwoga on 09/02/2023.
//

import SwiftUI


struct ContentView: View {
    //state vars for a,b,c,x1, and x2
    var body: some View {
        ScrollView {
            VStack {
              //  Text(...)
               // Label(...)
                //Image(...)
              //  TextField(...)
                // etc.

               // Button(...)
                }
            }
         //   Text(...)
        }

    }
  func solveForX()
    {
      //  your code here
    }



struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}
```

End of document