

# CSP Written Responses

Carson Angell

## What hardware requirements does this project need?

The only hardware needed to run this program would be an iPhone that is up to date with the latest software. However, to simulate or work on the program, you would need a Mac computer or some other computer that can develop in iOS.

## Explain this code block using "normal" English

```
78 struct MultiBasicShape : Shape {
79     func path(in rect: CGRect) -> Path {
98         return path
99     }
100
101     func makeBasicShape(in rect: CGRect, position pos: CGPoint) -> Path {
102         var path = Path()
103         var xPos = pos.x
104         var yPos = pos.y
105
106         path.move(to: CGPoint(x: xPos, y: yPos))
107         xPos += 70
108
109         path.addLine(to: CGPoint(x: xPos, y: yPos))
110         xPos -= 20
111
112         path.addArc(
113             center: CGPoint(x: xPos, y: yPos),
114             radius: 20,
115             startAngle: Angle(degrees: 0),
116             endAngle: Angle(degrees: -90),
117             clockwise: true
118         )
119         xPos -= 30
120         yPos -= 20
121
122         path.addArc(
123             center: CGPoint(x: xPos, y: yPos),
124             radius: 30,
125             startAngle: Angle(degrees: 0),
126             endAngle: Angle(degrees: -45),
127             clockwise: true
128         )
129         xPos += 20
130         yPos -= 20
131
132         path.addLine(to: CGPoint(x: xPos, y: yPos))
133         xPos -= 10
134         yPos += 10
135
136         path.addLine(to: CGPoint(x: xPos, y: yPos))
137         xPos -= 10
138
139         path.addLine(to: CGPoint(x: xPos, y: yPos))
140
141         path.addArc(
142             center: CGPoint(x: xPos, y: yPos),
143             radius: 10,
144             startAngle: Angle(degrees: 0),
145             endAngle: Angle(degrees: 135),
146             clockwise: true
147         )
148
149         path.closeSubpath()
150         return path
151     }
152 }
```



This method creates the little shape shown on the right. It starts on the bottom left. It moves to the right, creates those first two arcs upward to the left, and makes that little line that goes down and left. Then it draws that half circle on the top left and returns back to its original location with a line.

How would you help a friend fix an error in this type of code?

```
25 struct Triangle : Shape {
26     func path(in rect: CGRect) -> Path {
27         var path = Path()
28
29         path.move(to: CGPoint(x: rect.minX, y: rect.minY))
30         path.addLine(to: CGPoint(x: rect.maxX / 3, y: rect.midY))
31         path.addLine(to: CGPoint(x: rect.minX, y: rect.maxY))
32         path.closeSubpath()
33
34         return path
35     }
36 }
```

In this struct, one error that can occur is misspelling the property names of rect. So instead of saying rect.minY, it could be misspelled to rect.midY and vice-versa. If a friend encountered this type of error I would simply point out the spelling mistake and help them with any other errors they may have with this kind of code.

Write the code to iterate over all the items in this list in a different way.

```
29
30         Section("Artwork", isExpanded: $isShowingArtworks) {
31             ForEach(loadArtworks()){ artwork in
32                 NavigationLink(artwork.title, destination: ArtifactView(artifact: artwork))
33                     .accessibilityLabel("Link for \(artwork.title)")
34                     .accessibilityValue(artwork.title)
35             }
36         }
37     }
```

This iteration can be rewritten to use indices instead of a traditional for-each loop structure like so:

```
11     let artwork : [Artifact] = loadArtworks()
12
13     Section("Artwork", isExpanded: $isShowingArtworks){
14         ForEach(artwork.indices, id: \.self){ index in
15             NavigationLink(artwork[index].title, destination: ArtifactView(artifact:
16                 artwork[index]))
17                 .accessibilityLabel("Link for \(artwork[index].title)")
18                 .accessibilityValue(artwork[index].title)
19         }
20     }
```