Apple.App-Development-with-Swift-Certified-User.v2026-06-27.q14

□□□□:	App-Development-with-Swift-Certified-User
□□□□:	App Development with Swift Certified User Exam
□□□:	Apple
□□ □□ □□□:	14
□□:	v2026-06-27
# □□ □:	117
# □□ □□□:	140
https://www.krdump.com/Apple.App-Development-with-Swift-Certified-User.v2026-06-27.q14.html	

NEW QUESTION: 1

BlueView□ NavigationStack□ □□□□ RedView□ □□ □□□□ □□□□ □□□ □□□□□.

□ □□ □□□ □□□□ □□□□□.

Answer:

NavigationLink, .sheet

□□:

□ □□□ SwiftUI□ □□□ □ □□, □□ □□□□□ □□, □□, □□□ □□□□ □□□ □ □□□ □□□□□. □ □□ □□□ NavigationLink□□ □□□. SwiftUI□ NavigationStack □□□ □□□□□ □□□ □□□□ □□□□□ □□ □□□ □□□ □□ □□ □□□□□ □□□□ □□□□□. Apple □□□ □□□ □□□□ NavigationLink□ □□□□ □□□□ NavigationStack □□ NavigationSplitView □□□ □□ □□ □□□□□. □□ □ □□ □□ □□, □ "□□□ □ □□"□ □□□ BlueView()□ □□□□ □□□ □□□ □□□□□.

□ □□ □□□ .sheet□□ □□□. □□□□ □□□□ isPresented: $showRedView□ □□□□□, □□ □□ □□□□□ □□□□ □□ □□□ □□ □□ SwiftUI □□ □□□□□ □□□□□. Apple □□□ □□□, □□ □□ true□ □ □ □□ □□ □□□□□ sheet(isPresented:onDismiss:content:) □□□□ □□□□ □□□. □□□ showRedView□ □□□□□ SwiftUI□ RedView()□ □□ □□□ □□ □□□□ □□□□□.
□□□ □□□ □□□ □□□□□ □□□ □□□□.

```
NavigationLink( "□□□ □□ □□" ) {
□□□()
}
sheet(isPresented: $showRedView) {
□□□()
}
```

□□ Swift□ □□□ □ □□ □□ □□□□ SwiftUI □□ □ □□ □□□□□□ □□□ □□□□□ □□□□□.

```
1  struct ContentView: View {
2      @State var showRedView = false
3      var body: some View {
4          NavigationStack{
5              navigationLink("Show Blue View") {
6                  BlueView()
7              }
8              Button("Show Red View") {
9                  showRedView.toggle()
10             }
11         }
12         sheet(isPresented: $showRedView, content: {
13             RedView()
14         })
15     }
16 }
```

NEW QUESTION: 2
□□□ □□□ □□□□ □□□ □□ □□□□□□.

```
1  Text("That's all folks")
2    .padding(.top, 80)
3    .padding(.horizontal)
4    .foregroundColor(.green)
5    .font(.title)
6    .fontWeight(.heavy)
```

□□□ □□□ □□□□ □□ □□ □□□□ □□ □□□□ □□□ □□□□□□ □□□□□□.
□□□□ □□□□ □□ □□ □□□ □□□□□.
ExtractedView□ BigGreenTextView□ □□□□□□□.
□□□ □□ □□ □□ `let text: String` □□ □□□□□.
"That's all folks" □□□ □□□ □□□ □□□ □□□.
□□ □□□ □□□ □□ □□□ □□□ □□□□ □□□?

A. BigGreenTextView("□□ □□□□□ □□□")
B. BigGreenTextView(text: " That's all folks ")
C. BigGreenTextView(text: □□□)
D. BigGreenTextView(□□□)
Answer: B (LEAVE A REPLY)
□ □□□ SwiftUI□ □□□ □ □□, □□ □□□□ □ □□ □□□ □□□□□ □□ □□ □□ □□□□ □□□ □□□□.
SwiftUI □□□ □□ □□ □□□ □□□ □□ □□ □□□ □□□ □:
let text: □□□
□□ □□□ □□ □□□ □□□□ □□□□□. □□ □□□ 'text'□□□ Swift□ □□ □□□ □ `label text:` □□□□□ □□□□ □□ □□□□□ □□□□ □. □□□ □□□ □□□ □□□ □□□□.
BigGreenTextView(text: "□□ □□□□□ □□□")
□□□ B□ □□□□□.
□□ □□□□ □□ □□:
* A□ □□□□ □□□□ □□□□ □□□□ □□□ □□□ □□□□□□□.
* C□ □□ □□□ □□ text□□ □□□ □□□ □□, □□□ □□□ □□□ □□□ □□ □□ □□□ □□□□□ □□□□ □□□ □□□□.
* D□ □□□□ □□□□ □□□□□□□ □□□ □□□□.
□□ □□□□ SwiftUI □□□□□. □□□ □□□ □□□ □□□ □□□ □□ □□ □□□□, □□ □□□ □□□ □□, □□ □□□□ □□□□ □□□ □□ □□□ □□□□ □□ □□□□ □□□□□.

NEW QUESTION: 3
□□□ □□□□□:

```
let animals = [ Animal( id: 1, name: "Cat"), Animal( id: 2, name: "Dog"), Animal(id: 3, name: "Rabbit") ]
struct ContentView: View {
    var body: some View {
        List(animals) {animal in
            // <Add code here> //
        }
    }
}
```

Animal□□□ □□□□ □□□□ □, □□□ □□□ □□□□□ 5□□ □□ □□ □□□ □□□□ □□□?

A. □□□(□□ □□)
B. □□□(Animals[animal].name)
C. □□□(animals[animal].name)
D. □□□(□□ □□)

Answer: D (LEAVE A REPLY)
□ □□□ SwiftUI□ □□□ □ □□, □□ □□□ □□ □□□□ □□□□ □□□□ SwiftUI□□ □□ □□□□ □□□□ □□ □□□ □□□□□. □□□□ animals□ List(animals) { animal in ... } □□□□ □□□□ □□□ □□□□□. □ □□□□ □□□□ □ □□□ □□□ □□□□, □□□□ animal□ □□ □ □ □□ Animal □□□□□ □□□□□. UI□ □□ □□□ □□□ □□□□□ Text(animal.name)□ □□□□ □□ □□□ □□□□□. SwiftUI□ □□□ □ □□ □□□ □□□□ □ □□ SwiftUI □□□ □□, Text □□ □□□□ □ □□□ □□ □□□ □□ □□□□ □□□□ □□□□□.
□□□ □□ D□ □□□□□. □□□□ List □□□□□ □□ □□ □□ □□ □□□ name □□□ □□□□ □□□□□. □□ □□ □□□ □□ □□□ □□ □ □ □□□□ □□ SwiftUI □□□□□. □, □□□□ List□ □□□□, □□□□□ □ □□□ □□ □□ □□□ □□□□□□ □ □□ □□□□ □□□□.
Apple□ SwiftUI □□□□□□□ □□□ □□□ □□□ □□□ □□ □ □□ □□□ □□□□□ □□□□□.
□□ □□□□□ □□□ □□□ □□□□□□□. A□ □□ □□□□ □□ Animal□□□ □□ □□□ □□ □□□□ □□□□. B□ □□□ □□□□ □□□ □ □□ □□□ □□□□ Animals□ □□□□ □□□□. C □□ animal□ □□ □□ □□□ □□ □□□□ □□□□ □□□□. □□□□ □□ □□ Animal□ □□□□□ animal.name□ □□□□ □□ □□□ □□ □□□ □ □□□□.

NEW QUESTION: 4
□□□ □□□□□.

```
struct ContentView: View {
  var body: some View {
    Capsule()
      .fill(.blue)
      .frame(width: 200.0, height: 100.0)
      .[          ](
        [          ]("Great!")
          .font(.largeTitle)
      )
  }
}
```


□□ □□□ "Great!"□□□ □□□ □□□□ □□□.
□ □□ □□□ □□□□ □□□□□.
Answer:
□□□□, □□□
□□:
□ □□□ SwiftUI□ □□□ □ □□, □□ □□ □□ □□□□ □□□□ □□ SwiftUI □□ □□ □□ □/□□ □□□□□ □□□ □□□ □□□□. □□□ □ □ □□ □□ □□□□ □□□□□ SwiftUI□□□ □□□□ □□□□ □□□□□. Apple □□□ □□□ □□□□□ □□□ □□ □□ □ □□ □□ □□ □ □□□□□, □□ □□ □ □□□ □□□ □□□□□. □□□□ □□□ □□ □□□ □□□ □□ □□ □□□□ □□□. □□□ □ □□ □□□□ Text□ □□ □□ □□□. SwiftUI□ "Great!"□ □□ □□□ □□□□ □□□□ □□ Text □□ □□□□ □□□□□.
□□□ □□□ □□□ □□□□.

```
struct ContentView: View {
var body: □□ □ {
□□()
.fill(.blue)
.frame(width: 200.0, height: 100.0)
.□□□(
□□□("□□□□!")
.font(.largeTitle)
```

)
}
}
□ □□□ □□□□ □□□ `Capsule()` □□□ □□□ □□□□, `.fill(.blue)`□ □□□□ □□□, `.frame(width:height:)`□ □□□ □□□□, `.overlay(...)`□ □□□□ □□□("Great!")□ □□ □□ □□ □□□□ □□□□□. □□ SwiftUI□ □□ □□ □□□□□. □□ □□ □□ □□, □□□□ □□□□ □□□□ □□□□ □□ □□□□ □□ □□□□ □□□□□. Swift□ □□□ □ □□ □□□□ □□ SwiftUI□ □□ □, □□□ □ □□□□ □□□ □□□□ □□ □ □□□□□□.

NEW QUESTION: 5
□□ □□ □□□ □□□□ □□□□□ □□□ □ □□ □□ □□□□□ □□□□□□.

A. "□□", "□□", "□□□", "□□□"□ □□□ □□ □□□ □□□□□.
B. "□□", "□□", "□□", "□□□", "□□□"□ □□□ □□ □□□ □□□□□.
C. "□□", "□□", "□□□", "□□□", "□□□□"□ □□□ □□ □□□ □□□□□.
D. for □□□ □□□ □□ □□ □□ □□□ □□□□□.
Answer: A (LEAVE A REPLY)
□ □□□ Swift □□□□□ □□, □□ □□□ □□□ □□ □□ □□□ □□□□□.
□□□ □□ □□□□ □□□□□. □□□□ 6□□ □□□ □□□□ menuItems.count□ 6□□□. □□□ □□□ □□ □□□□ □□□□□.
□□□□ 1□□ 1□□□ □□, □□ □□ □(menuItems.count)□□ □□ □(menuItems.count - 1)□□ □□□□□.
□□□ □□□ □□ □□□:
□□□□ 1.. < 5□ □□
Swift□□ □□□□ □□ □□□ a.. < b□ □□□ □□□□□ □□□ □□□□ □□□□.
□□□ □ □□□ □□□ □ 0□□ 5□ □□ 1, 2, 3, 4□ □□□□□.
□□ □□□□ □□ □□□□ □□□□ □□□□□.
* menuItems[1] # "□□"
* menuItems[2] # "□□"
* menuItems[3] # "□□□□"
* menuItems[4] # "□□□"

□, □□□ 0□ □□ "Pizza"□ □□□ 5□ □□ "Steak"□ □□□□ □□□. Swift □□□ 0□□ □□□□ □□□□ □□□□, `count`□ □□□ □□ □□□ □□□□□.
□□□ □□□□□ "□□", "□□", "□□□", "□□□"□ □□□□□ □□□ A□□□.

NEW QUESTION: 6
□□ □□□□ □□□ □□□□ `self` □□□□ □□□ □□□ □□□□□?

A. self□ □□ □□□□□ □□□□ □□□ □□□□ □□□ □□ □□□□□.
B. □□□ □□□□ □□□ □□□ □□ □□ self□ □□□□□.
C. self□ □□□ □□□ □□ □□□ □□□□□ □□□□ □ □□□□□.
D. self□ □□□ □□□□ □□ □□□□ □□□□□.
Answer: (SHOW ANSWER)
□ □□□ Swift □□□□□ □□, □□ □□□, □□ □ □□□ □□□□ □□□ □□ □□□ □□□□□.
□□ □□□□□ □□□ □□□ □□□□ □□ age□□ □□□ □□□□□.
struct person {
□□ □□: □□
init(age: Int) {
self.age = □□
}
}
□□□ □□□ □□□ □□ `age`□ □□□□ □□ □□ □□□ □□□□ □□□ □□□□□ □□□ □ □□□□. Swift□□□ `self.age`□ □□□□ □□ □ □□□□ □□□ □□□ □□□□ □□□□ □□, `age`□ □□□ □□□□ □□□□□ □□□□□. □□□ `self.age = age`□ □□□□ □□ □□□□□ □ □□ □□□□□ □□□□□.
□□□ C□ □□□□□. □□□ □□□ □□ □ □□□ □□ □□□□ □□□ □□ □□□□ `self` □□□□ □□□□□.
□□□ □□□□ □□□□□.
* A□ □□□□□. □□□ self□ □□□□□ □□□□ □□ □□□ □□□□ □□□ □□□□ □□□□□.
* B□ □□ □□□□□. □□ □□□ □□□ self□ □□ □□□ □□ □□□□. □□□□ □□ □□ □□□ □□□ □□□ □□□□.
* D□ □□□□□. □□ □□□ □□□□ □□□ □□□ □ □□□□ self□ □□□ □□□ □□□□.
□□□ □□□ C□□□. self□ □□□ □□□ □□ □□□ □□□□□ □□□□ □ □□□□□.

NEW QUESTION: 7
SwiftUI□□ □□ □□□ □□ □□ □□□□ □□□ □□ □□□□□□ □□□ □□ □□□?
A. □□□□ □□ □ □□□ □□□ □□□□ □□ □□□□□.
B. □□□□ □□□ □□ □□ □□□ □□ □□□□□.
C. □□□ SwiftUI □ □□□□ □□□□ □□ □□□ □□□□□.
D. @State□ □□□□ □□□ □□□□ □□□□ □□□□ □□□□ □□□ □□□ □□□□□.
Answer: C (LEAVE A REPLY)

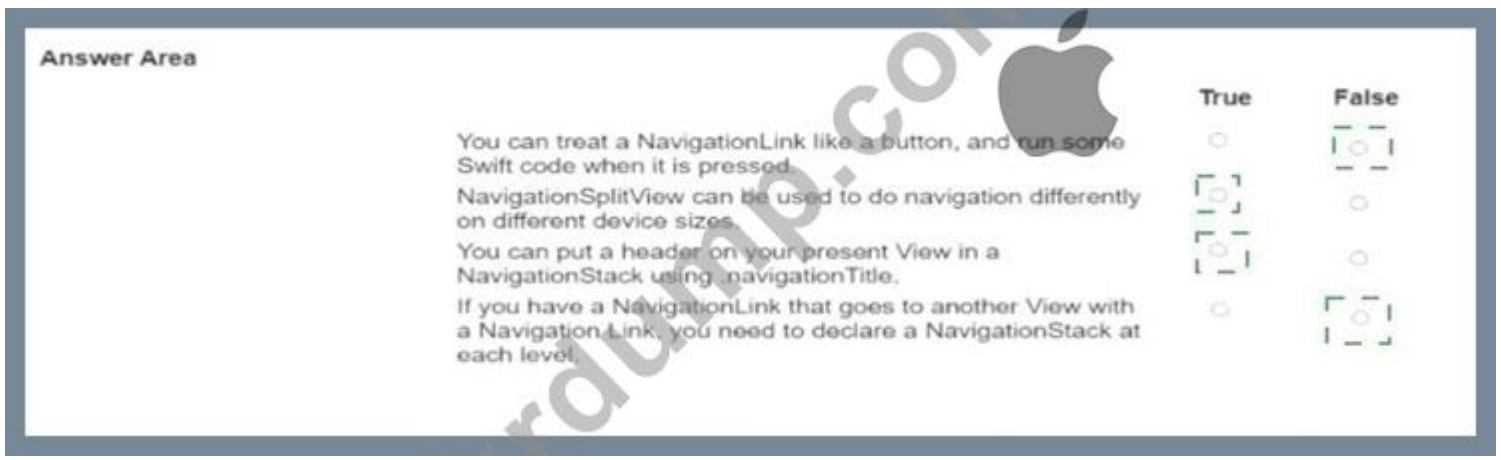
Swift□ □□□ □ □□ □□□ □□ □□□□□ □□□ □□:
□ □□□ SwiftUI□ □□□ □ □□, □□ □□□□ □ □□ □□□ □□□□□ □□ □□ □□ □□□□ □□□ □□□□□. □□□ C□□□. SwiftUI□□ □ □□□□□□ □□□□□ □□□□ □□ □□□ □□□ □□□ □□ □(□□□□□ View □□□□□□ □□□□ □□□ □□□)□ □□□, □ □□ □□ □ □□□ □□□□ □□□ □□□□□. Apple□ □□□□□ □□□□□ □□□ □□□ □□□□□ □□□□□. □, UI□ □□□ □□□ SwiftUI □ □□□□ □□□ □□, □ □□ □ □□ □□□□ □□□□ □□ □□□ □□□□ □□□□.
□□ A□ □□□ □□□ □□□□ □□□□ □ □□□□ □□ SwiftUI □□□ □□□□. □□ B□ □□ □□ □□□ □ □□□ □□□□ □□□ □□□□□ □□□□□□. □□ D□ □□ □□ □ □□□ □□□ □□ □□□, □□□ □□ □□□ □□□ □□□ □□□□ □□□□ □□ □□ □□ □□□□. Apple□ SwiftUI □□□ □□ □□□□□ □□□, □□ □□□ □ □□□□□ □□ □ □□ □□□ □□□ □□□□ □□□ □ □□□ □□ □□□ □□□□□.
WWDC □□□□□□ □□□ UI □□□□ □□□ □ □□□ □□□□ Xcode□ "□□□ □□" □□□□□ □□□□□.

NEW QUESTION: 8
□□□ □□□□□.
struct ContentView: View {
□□ □□ = ["□□" , "□□□" , "□□"]
var body: □□ □ {
List(fruits, id: \.self) { □□
□□□(□□)
.font(.headline)
.□()
}
}
}
□□ □ □□□ □□ □□□□ □□ □□ □□□□□?
A. □□□ □□□ id: \.self□ □□□ □□□□ □□□ □ □□□□.
B. □□□□ □□□□ □□□□ □□ □ □□ □□□ □□□□ □□□□□.
C. □□□ □□ id: \.self□ id: /self□ □□□□ □□□.
D. □□□ □□ fruits □□□ □□□□ □□□ □□ □□□ □□□□□.
Answer: (SHOW ANSWER)
Swift□ □□□ □ □□ □□□ □□ □□□□□ □□□ □□:
□ □□□ SwiftUI□ □□□ □ □□, □□ □□□□ □□□□ □□□ □ □□ □□□ □□□□. □□□□ fruits□ □□□ □□□□, □□□ □□□ □□□□ □□□□ □□ □□□□ □ □□□ □□ □□□ □□ □□□□ □□□□. Apple□ SwiftUI □□□ □□□ □□□□ □□□ □□□□□ □□ □□□ □ □□ □, □□□ □□□ □□ □□□□ □□□□ □□□□ □□□□ □□ □□ SwiftUI□ □ □□ □□□□ □□□ □ □□□ id □ □□□ □□□ □ □□□□. □ □□ `id: \.self`□ SwiftUI□□ □ □□□ □ □□□ □□□□ □□□□□ □□□□□.
□□□ D□□ □□□□□. List□ fruits □□□ □□□ □□□□ □□ □□□ □□□□□ □□□, □ □□□ Text(fruit) □□ □□□□□. Apple□ □ □□ □□ □□□□□ List□ □□□ □□□ □□ □□ □□□ □□ □□□□ □□□□ □□ □□□□ □□□, □□ □ □□□ □□□ □□□ □□□□□.
□□ A□ □□□□□. □□□□ □□□ □□□ □□□ □□□ □□ □ □□ □□□□ □□ id: \.self□ □□□□ □□□□□.
□□ B□ □ □□□ .font(.headline) □□ .padding()□ □□□ □□□□ □□□□□. □□□ □□ □ □□□□□, □ □□□□□ □□ □□ □□□ □□□□.
□□ C□ □□□□□. Swift□ □ □□ □□□ \.self□□ □□□□□ □□□□ /self□□ □□□□ □□□□. Apple□ KeyPath □□□ □□□ Swift □ □□□ □□□□ □□□ □□□□□.

NEW QUESTION: 9

SwiftUI□ □□□□□□ □□ □ □□□ □□ □ □□ □□□ □□□□□.

Answer:



□□:

* NavigationLink□ □□□□ □□□□ □□□ □ Swift □□□ □□□ □ □□□□. - □□

* NavigationSplitView□ □□ □□ □□□ □□ □□ □□□□ □□□ □ □□□ □□□□. - □□□□

NavigationStack□□ □□ View□ .navigationTitle□ □□□□ □□□ □□□ □ □□□□. - True

* NavigationLink□ □□ View□ □□□□ □□, □ □□□□ NavigationStack□ □□□□ □□□. - □□. □ □□□ SwiftUI□ □□□ □ □□, □□ □□□□ □ □□, □□ □ □□□ □□□□ □□ □ □□ □□□ □□□ □□□□□. □ □□ □□□ □□□□□. NavigationLink□ □□ □□□□□ □□□□□ □□ □□ □□□□□ □□□□ □□□□□ □□□□□□. Apple □□□ □□□ NavigationLink□ □□□□ □□□□ □□□□□ □□□ □□□□ □□□, Button□ □□ □□□□ □□ □□□□ □□□□.

□ □□ □□□ □□□□. NavigationSplitView□ □□□ □□□□□□ □□ □□□□□□ □□□□ □□ □□□ □□□ □□ □□□ □□□□ □□□□□□ □□□ □ □□ □□□□□. Apple□ NavigationSplitView□ □□ □□ □□ □□□□□□ □□□□ □□□□□ □□□□ □□□, □□□ □□□ □□ □□□ □□ □□□ □□ □□□ □□□□ □□□□.

.navigationTitle(...) □□□□ □□□□□ □□□□ □□□ □□□□ □□ □□□□□ □□□ □□□□ □□□ □ □□ □□□ □□□□. Apple□ □□ □□ □□□ □□□ □□□□□□ □□ □□□□□ □□□ □□□□□ □□□□ □ □□□□ □□□ □□□□□ □□□□□.

4□ □□□ □□□□□. □□ □□□ □□□ NavigationStack□ □□□ □□ □□□ □□□□□. Apple□ NavigationStack□ □ □□□ □□□□ □□□□□ □□□□, NavigationLink□ □□□□ □□□□ □□ □□□ □□□□□. □□□ □□□□ □□□ □□ □□□ □□ □□□ □ □□□□.

NEW QUESTION: 10

□□ □□ □□□ □□□□□.

```swift
struct Document {
    var content: String
    static var docCount = 0
    var description: String {
        return "Document Content:\(content)"
     }
    static func increment() {
        docCount += 1
    }
    func display() {
        print("Content:\(content)")
    }
}
let wordDoc = Document(content: "Greetings!")}
```

□□ □□□ □ □□□ □□□□ □□□ □□ □□□□□ □□□□□. □ □□□ □ □□ □□□ □ □□□□.
□□: □□□ □□□ □□ □□□ □□ □□□.

Answer:

□□:

□ □□□ Swift □□□□□ □□, □□ □□□, □□, □□□ □ □□□ □□□ □□□ □□□□.
□□ □□□ □□ □□ □□□□ □□□□. □□ □□ □□□□□□ □□□ □□ □□□□□.
□□□ □□□ description□ □□ □□□□□. □□□□ □□□□ □□□□ □□□□□.
Swift□ □□□□ □□ □□□ □□□□□ □□ □□□□ □ □□□ □□□ □□□□ □□□□ □□□□□. □□□ `Document(content: " Greetings!" )`□ □ □□□ □□□ □□□ □□□□ □□□□ □□□□.
□□ □□□ □□□□□ □□ □□ □□ □□□ □□□□. Swift□□ `static var docCount = 0`□ `static`□□ □□□□□ □□□ □□ □□□□□.
□□□□ □□□□ □□□ □□ □□□□ □□□□□ □□□ □□□□□. display() □□□□ □□□□□ □□□ □□□□□ □□□□ □□□□□□.
□□ □□□□ static□□ □□□□ □□ □□ □□□ □□□ □□□□□□. □□□ static func increment()□ □□ □□ □□□ docCount□ □□□□□ □□ □□□□□□.

NEW QUESTION: 11
□□ □□ □□□ □□□□□.

□ print □□ □□ □□□ □□□□□?
Answer:
□□□ □□ □□ □□□□ □□□□□.
10
□□:
□ □□□ Swift □□□□□ □□, □□ □□□, □□□, □□, □□□ □ □□□□ □□□□ □□□□ □□□ □□□ □□□□.
□□□ Printer□ □□□□ □□□□□□ □□□□.
□□□ Printer {

□□ □□□: □□
init(copies: Int) {
self.copies = □□□
}
}
Swift□□ □□□□ □□ □□□□□. □, □ □□□ □□□□□ □□ □□□ □□□□ □ □□□ □□□□□ □□□ □□□ □□□□, □□□ □□□□ □□ □□ □□□□. Apple□ Swift □□ □□□□□□ □□□□ □□□ □□□□ □□, □□□□ □ □□□□□ □□□□□. □□□ □□ □□□□:
var □□□1 = □□□(□□□: 2)
var printer2 = printer1
printer1□ printer2□ □□ □□□ Printer □□□□□ □□□□□.
□□□□, □ □□ □□ □□□ □□□□□.
printer2.copies = 10
printer2□ printer1□ □□□ □□□□□ □□□□□ printer2.copies□ □□□□ printer1□ □□□□□.
□□□□□□. □□□ □□□ □□□ □:
print(printer1.copies)
□□□□ 10□□□.
□ □□□ Swift□□ □□ □□□ □□ □ □□□ □□□□□□. □□□□ □□ □□□□ □□□□ □ □□□□□. □□ `Printer`□ □□□□ □□ □□□□□ □, □□ □□ □ □□□ □□□□□ □□□□ □□ □□ □□□□ □□□ □□□ □□□□ □□□□.

NEW QUESTION: 12
□ □□□□ □□□□ □□□ □□□ □□□□ □□ □□□ □□□□□ □□□ □□□□□.

□□: □□□ □□□ □□ □□□ □□ □□□.

Answer Area

```
import SwiftUI

structContentView: View {
    var body: some View {
        VStack {
            Text( "Press To Play")
                [ ▾ ]
                    Text
                    Label
                Button
                [ ▾ ] ("Ready, Set, Go!")
                    Label
                    String
                    Font
                    Text
                [ ▾ ] ( .largeTitle)
                    .font
                    .text
                    .label
                [ ▾ ] (15)
                    .space        ıe,width: 5)
                    .padding
                [ ▾ ] (.yellow)
            }) .color
                .backgroundColor
            }
                background
        }
            .textcolor
```

Answer:

Answer Area

```
import SwiftUI

structContentView: View {
    var body: some View {
        VStack {
            Text( "Press To Play")
                Text
                Label
                Button
                    ("Ready, Set, Go!")
                    Label
                    String
                    Font
                    Text
                        ( .largeTitle)
                        .font
                        .text
                        .label
                        (15)
                        .space        ıe,width: 5)
                        .padding
                        (.yellow)
                    }) .color
                .backgroundColor
                .background
                .textcolor
        }
    }
}
```


Answer Area

```
import SwiftUI

structContentView: View {
    var body: some View {
        VStack {
            Text( "Press To Play")
            Button  (action: {}, label: {
            Text    ("Ready, Set, Go!")
                .font   ( .largeTitle)
                .foregroundStyle( .blue)
                .padding    (15)
                .border(.blue,width: 5)
                .background     (.yellow)
            })
        }
    }
}
```


NEW QUESTION: 13

□□ □□ □□□ □□□□□.

```swift
let unitPrice = 2.5
let quantity = 20
let shipping = 9.9
let totalCost: Double = 0
totalCost = unitPrice * quantity + shipping
print(totalCost)
```

□□ □□ □□□ □□□□□ □□□□.
□□□ □□□ □ □□ □ □□ □□□ □□□□□? (□ □□□ □□□□□.)
A. □□□ □□□ □□□ int□□ Double□ □□□□□.
B. totalCost□ □□□□ 0□□ 0.0□□ □□□□□.
C. unitPrice □□□ Double□□ Int□ □□□□□.
D. totalCost□ let□□ var□ □□□□ □□□□□ □□□□.
E. shipping □□□ let□□ var□ □□□□ □□□□□ □□□□.
Answer: A,D (LEAVE A REPLY)
□ □□□ Swift □□□□□ □□, □□ Swift□ □□ □□□ □□□ □□□ □□□ □□□□.
□□□, □□□ □□□ □□□.
□□ □□ □□□□ □ □□ □□□ □□□ □□□□. □□, unitPrice□ shipping□ Double□ □□□□ □□ quantity□ Int□ □□□□□. Swift□□□ □□ □ □ □□□□□ □□□ □□□□□ □□□. Swift□ □□□ □□ □□□□□ Int□ Double□ □□□□ □□□□ □□□□. □□□ quantity□ Double□ □□□ □□ □□□□□ □□□□ □□□ unitPrice * quantity □□□ □□□□□. □□□ A□ □□□ □□ □□□□□.
□□, `totalCost += ...` □□ □□ □□ □□□ `+=`□ □□□□□, □ □□□□ □□□ □□□ □□ □□□□□. Swift□□□ `+=` □□□□ □□□ □□ □□ □□ □□□ `totalCost`□ `let`□ □□ `var`□ □□□□ □□□. □□□ D□ □ □□ □□□ □□□□□.
□□ □□□□□ □□ □□□ □□□ □□□□ □□□□. B□ totalCost□ □□ Double□□□ □□□□□ □□□□ □□□□ 0□ □□□□ □□□ □□□□ □□. C□ shipping□ □□□ Double□□□ □□□□□ □□□ □□ □□□ □□ □□□□. E□ shipping□ □□□□□ □□□ □□□□ □□□ □□□□.
□□□ □□□ A□ D□□□.

NEW QUESTION: 14
□□□ □□□ □□, □□□ □□ □□ □ □□□ □□□□□? (□ □□□ □□□□□□.)

```swift
func rightSum(_ num1: Int, by num2: Int, and num3: Int = 25) -> Int {
    return num1 + num2 + num3
}
```

A. let sum = rightSum(100, num2: SO, and: 20)
B. let sum - rightSum(100, by: 30, and: 20)
C. let Sun - rightSum(numl: 100, by: SO, and: SO)
D. let sum r\ ght5um(100, by: 50)
E. sun = rightSum(numl: 100, num2: 50)
Answer: B,D (LEAVE A REPLY)
□ □□□ Swift □□□□□ □□, □□ □□ □□(□□ □ □□ □□□□ □□, □□ □□□□ □ □□)□ □□□□.
□□□ □□□ □□ □□□□□.
func rightSum (_num1:Int, by num2:Int □ num3:Int = 25) -> Int {;
num1 + num2 + num3□ □□□□□.

}
□□ □□□ □□□□□.
* □ □□ □□□□□ □□(_)□ □□□□□ □□ □□□□ □□□□.
* □ □□ □□□□□ □□ □□□□ □□□□ □□□.
* □ □□ □□□□□ □□ □□□□ □□□□ □□□.
* □ □□ □□□□□ □□□□ 25□□□ □□□ □ □□□□. □□ □ □□□ □□□ □□□□□.
* A□ □□ □□ □□□□ by □□ num2:□ □□□□ □□□ □□□□ □□□□.
* B□ □ □□ □□□ □□□□ □□□□ □□ □□□□ □□□□ □□□ □□□□□. □□□ □□□ □□ □□□ □□□□□.
* C□ □ □□ □□□□□ num1□□ □□□ □ □□□□ □□□□ □□□□. □□ □□□□ _□ □□□□□ □□□□□.
* D□ □ □□ □□□ □□□ □□, □ □□ □□□ by:□ □□□□ □□□□ □ □□ □□□ □□□□ Swift□ □□□ 25□ □□□□□ □□□ □□□□□.
* E□ □□ □□ □□ □□ num1: □ num2:□ □□□□ □□□ □□□□ □□□□. □□□ □□□ □□ □□□ B□ D□□□.