Assessment Answers:

# Q3

1. Protocol oriented
2. Optionals
3. Open source
4. Efficient memory management
5. Can be used alongside Objective C and SwiftUI

# Q6

The minimal components required are:

1. UIViewController
2. UITableViewCell
3. UITableViewDataSource

# Q8

The value for score1.score would still be 5. It is because when we do score2 = score1, it will create a new instance which is a copy of score1 and hence updating the value of score2.score does not affect score1.

# Q9

The correct statements are:

1. The initializer init is required to set the initial values of all properties.
2. The property “name” cannot be changed after initialization.
3. The “mutating” keyword of function “changeAddress” is required to modify the instance variable “address”.

# Q10

1. ? (Optional): It is used to create optional variables. Optional variables are variables which can have nil value.
2. ?? (Nil – coalescing operator): It is used to assign default values if the value in the optional variable is nil.
3. It can wither be:
4. ! (Not): To inverse the Boolean value
5. ! (Force unwrapping): It is used to forcefully unwrap the value of an optional variable. If the value is nil, this will cause the app to crash.
6. ..< (Range to include the lower bound but exclude the upper bound): This will create a range that will start from the lower bound, i.e., the character before the operator, and will go till the value of 1 less than the upper bound.
7. … (Range to include lower and upper bound both): This will create a range that will start from the lower bound and go till the value of the upper bound.

# Q14

The problem with this line of code is that we are force unwrapping the code. This is cause an error if there is nothing stored in the user defaults for the particular key. We can make it better by unwraping the optional code using any of the 3 ways mentioned in Q5. Here’s an example code:

**var** defaults = UserDefaults.standard

**var** name = defaults.string(forKey: "name") ?? "Cannot find name"

printString(string: name)

**func** printString(string: String) {

print(string)

}