
Guru99 Provides [FREE ONLINE TUTORIAL](#) on Various courses like

[Java](#) | [MIS](#) | [MongoDB](#) | [BigData](#) | [Cassandra](#) | [Web Services](#)

[SQLite](#) | [JSP](#) | [Informatica](#) | [Accounting](#) | [SAP Training](#) | [Python](#)

[Excel](#) | [ASP Net](#) | [HBase](#) | [Testing](#) | [Selenium](#) | [CCNA](#) | [NodeJS](#)

[TensorFlow](#) | [Data Warehouse](#) | [R Programming](#) | [Live Projects](#) | [DevOps](#)

Top 15 Swift Interview Questions & Answers

1) Explain what is Swift Programming Language?

Swift is a programming language and system for creating applications for iOS and OS X. It is an innovative programming language for Cocoa and Cocoa Touch.

2) Explain how you define variables in Swift language?

Variables and constants must be declared before they are used. You announce constants with the let keyword and variables with the var keyword. Both variables and dictionaries are described using brackets. For example,

```
Var Guru99 = "This is Guru99"
```

```
Let ksomeconstant = 30
```

3) What is the significance of "?" in swift?

The question mark makes a property optional if declared. In case the property does not hold a value, the "?" helps in avoiding runtime errors.

4) Mention what are the features of Swift Programming?

- It eliminates entire classes of unsafe code
- Variables are always initialized before use
- Arrays and integers are checked for overflow
- Memory is managed automatically
- Instead of using "if" statement in conditional programming, swift has "switch" function

5) Mention what is the difference between Swift and 'Objective-C' language?

Difference between 'C' and 'Swift' language is that

Swift

- In a swift, the variable and constants are declared before their use
- You have to use "let" keyword for constant and "var" keyword for variable
- There is no need to end code with semi-colon
- Concatenating strings is easy in swift and allows to make a new string from a mix of constants, literals, variables, as well as expressions
- Swift does not require to create a separate interface like Objective C. You can define classes in a single file (.swift)
- Swift enables you to define methods in class, structure or enumeration
- In Swift, you use " += " Operator to add an item

Objective-C

- In objective C, you have to declare variable as NSString and constant as int
- In objective C, variable is declared as " " and constant as " "
- The code ends with semi-colon
- In objective C, you have to choose between NSMutableString and NSString for string to be modified.
- For classes, you create separate interface (.h) and implementation (.m) files for classes
- Objective does not allow this
- In C, you use "addObject": method of NSMutableArray array to append a new item to an array



6) Mention what are the type of integers does Swift have?

Swift provides unsigned and signed integers in 8, 16, 32 and 64 bit forms. Similar to C these integers follow a naming convention. For instance, unsigned integer is denoted by type UInt8 while 32 bit signed integer will be denoted by type Int32.

7) Mention what is the Floating point numbers and what are the types of floating number in Swift?

Floating numbers are numbers with a fractional component, like 3.25169 and -238.21. Floating point types can represent a wider range of values than integer types. There are two signed floating point number

- Double: It represents a 64 bit floating point number, it is used when floating point values must be very large
- Float: It represents a 32 bit floating point number, it is used when floating point values does not need 64 bit precision

8) Explain how multiple line comment can be written in swift?

Multiple line comment can be written as forward-slash followed by an asterisk (/*) and end with an asterisk followed by a forward slash (*/).

9) What is de-initializer and how it is written in Swift?

A de-initializer is declared immediately before a class instance is de-allocated. You write de-initializer with the deinit keyword. De-initializer is written without any parenthesis, and it does not take any parameters. It is written as

```
deinit {  
  
    // perform the deinitialization  
  
}
```

10) Mention what are the collection types available in Swift?

In Swift, collection types come in two varieties Array and Dictionary

- **Array:** You can create an Array of a single type or an array with multiple types. Swift usually prefers the former one

Example for single type array is,

```
Var cardName : [String] = [ "Robert" , "Lisa" , "Kevin"]
```

```
// Swift can infer [String] so we can also write it as:
```

```
Var cardNames = [ "Robert", "Lisa", "Kevin"] // inferred as [String]
```

To add an array you need to use the subscript `println(CardNames[0])`

- **Dictionary:** It is similar to a Hash table as in other programming language. A dictionary enables you to store key-value pairs and access the value by providing the key

```
var cards = [ "Robert": 22, "Lisa" : 24, and "Kevin": 26]
```

11) List out what are the control transfer statements used in Swift?

Control transfer statements used in Swift includes

- Continue
- Break
- Fallthrough
- Return

12) Explain what is optional chaining?

Optional chaining is a process of querying and calling properties. Multiple queries can be chained together, and if any link in the chain is nil then, the entire chain fails.

13) How base-class is defined in Swift?

In Swift the classes are not inherited from the base class and the classes that you define without specifying its superclass, automatically becomes the base-class.

14) Explain what Lazy stored properties is and when it is useful?

Lazy stored properties are used for a property whose initial values is not calculated until the first time it is used. You can declare a lazy stored property by writing the lazy modifier before its declaration. Lazy properties are useful when the initial value for a property is reliant on outside factors whose values are unknown.

15) Mention what is the characteristics of Switch in Swift?

- It supports any kind of data, and not only synchronize but also checks for equality
- When a case is matched in switch, the program exists from the switch case and does not continue checking next cases. So you don't need to explicitly break out the switch at the end of case
- Switch statement must be exhaustive, which means that you have to cover all possible values for your variable
- There is no fallthrough in switch statements and therefore break is not required