

By OnlineInterviewQuestions.com

Haskell Interview Questions

In today's tough competition it's quite difficult to clear any interview. **Haskell** is one of the advanced programming languages, which provide easy integration with other languages, built-in concurrency, and rich libraries. It is perfectly suited for data analysis and proprietary business logic, enhancing current software with correct codification and scalability and performance.

There are various Haskell jobs in the market but to get selected in the job you need to clear the interview. Questions difficulty level can be different for fresher and experienced one. Freshers may get easy and simple questions whereas experienced candidates have to go through higher-level questions. So whether you are experienced or fresher, you need to prepare for the interview.

Having an interview with Haskell programming and thinking about the questions, which can be asked by you during the interview, come on we are here to sort out this issue for you. Go through below given questions, which might help you during the interview.

Q1. Define Haskell?

Haskell is the advanced functional programming language. Haskell is the language that is built around functions, using blocks of various codes that do specific tasks. It has a major benefit like shorter development time, cleaner code, and high reliability. It gives all the features like sketched above, including polymorphic typing, lazy evaluation, and higher-order functions.

Q2. In which language is Haskell written?

Haskell is written in various languages. It is a general-purpose programming language; Haskell is formed without having any application niche in mind. While at its core the language encourages a pure, a lazy style of functional programming. For example, GHC is

- **Runtime:** a combination of C, assembly, and C
- **Compiler:** Haskell
- **Libraries:** Haskell and some of C, a little assembly.

Q3. Define list? What are the ways to create a list in Haskell?

A list is a single linked list like one looks in an imperative language, but there is one major important difference. Only and the only operation we have is to put a node at the start of the list. We cannot change any values within a list, including the pointers from one list node to another.

Usage of square brackets can create a list in Haskell with commas, which separates the list's values. There are only four functions, which are normally required, but because lists are so widely used in Haskell programs, the prelude provides various another functions also.

Q4. State the key features of Haskell programming Language?

Various features of Haskell programming language are

- Lazy evaluation
- Statically Typed
- Type inference
- Purely functional
- Concurrent

Haskell is completely functional language, or another term would be mathematical which means that functions have no side effects. With different features, this language provides functional programming, which offers extreme small and compact method for coding programs.

Q5. What is functional programming?

C, Pascal, Java, and various more all are imperative languages whereas Haskell is a functional language. Functional programming is a single expression, which is executed by evaluating the expression. Anyone who has used a spreadsheet has experience of functional programming. Functional programming languages take the similar ideas and get them into the realm field of general-purpose programming.

Q6. Differentiate between Functional vs. imperative Programming languages?

Functional Programming is used in various languages like Haskell, Lisp, and scala. Whereas imperative programming is the most dominant paradigm of all the others. It is utilized in Java, C, python, ruby and much more. One of the major difference between imperative and functional languages is that imperative programming language functions that are evaluated are the main reason of gaining and transforming data, whereas functional is stateless. Most of the people are used to recursion in imperative languages, and functional languages could have special support for recursion.

Q7. What do you mean by higher-order functions?

A higher order function is that which takes a function as an alteration or returns a function. A higher order function accepts functions as arguments and can return a function as its result. It supports high order functions and allows functions to be constituents of data structures.

Q8. How to import a module in Haskell?

The following are the ways to import a module in Haskell.

- **Import the functions, which you need:** The first thing, which makes your import clearer, is to specify which functions you want to import. The biggest provocation is only to write down the module name in

the import. Once you have built a Haskell program of even modest size, then you will surely appreciate the importance of the import section of any source file.

- **Qualifying Imports:** After specifying the functions, which you need to import then move on the next way to clarify your imports is to use the qualified keywords. This clearly means that you should prefix every function you use from this module by the name assigned to the module.
- **Organizing your imports:** After qualifying imports, then next step is to organize the internal imports from external ones. So which means two list needs to create in the first list will consist of built-in-packages, and 2nd list has modules that are in the codebase itself.
- **Creating the list easy to read -:** At last you want to make it easy to read your import list. Write your list in alphabetical order, which helps people find the right module in the list.

Q9. What do you mean by Lambdas in Haskell?

It is because Haskell is a functional programming language, so one would assume functions to play a major role but indeed it does. So the use of lambdas in Haskell language is that instead of using equations to define various functions, we can also explain them “anonymously” through a lambda expression.

Q10. What do you mean by Functor Class in Haskell?

A functor class is simply something that can be mapped over. Functor transforms one category into another. Functor class serves as adapters between categories that promote code written for the source category to be automatically accessible with the destination category.

Please Visit OnlineInterviewquestions.com to download more pdfs