

What is `()` ?

What are Haskell's capitalization rules?

What is the key benefit of type variables over generics?

From an object-oriented point of view, a
typeclass can be thought of as ...

From Haskell's point of view an operator is different from a function only in that ...

What does `=>` do in a type signature?

What is Eq ?

What are some prominent non- Eq s?

Describe Haskell's equivalent of Java's
Comparable.

What is the `Show` typeclass?

How do you go back from `show`'s string representation to the object itself?

What's the catch?

What is a type annotation?

Members of `Enum` can do what?

What about `Bounded`?

Give two examples of polymorphic constants.

Make integral and floating point types play nice together.

What is that function's signature?

How can repeated computations be avoided
in a series of guards?

What additional kind of assignment can be made with `where/let`, pattern matching aside?

Where are helper functions typically defined,
when scope-limiting is desired?

Other than order, what's the difference
between `let/in` and `where`?

How can multiple `let` bindings be put on one line?

In addition to guards and generators, what
can go in list comprehensions?

Pattern matching on function parameters is
just syntactic sugar for ...

What is the full syntax?