

Describe `hashCode ()` contract.

When is there no need to create `canEqual` method?

Why not force `equals()` symmetry by checking the equality of runtime classes?

Other than Java interop, what is a common use of existential types?

Give the `hashCode ()` recipe.

What is your `hashCode()` only good as?

Since ____ is not great by default, what should you use?

Why is 41 used in `hashCode` recipe?

What are the equivalence relation `equals()` requirements for non-null objects?

Give some common `equals()` pitfalls.

Give the `equals()` recipe.

What should you do if your `hashCode()`
invokes `super.hashCode()` ?

Describe efficient `hashCode`.

What is a common source of `equals()` methods that are not equivalence relations?

What's an apparent inconsistency with Java's
`equals()` ?

What is `Any's ==` defined as?