

These differences are based on Java 7:

| Interface | Abstract Class |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Methods of an interface are abstract and cannot have implementations | Abstract class can have instance methods with default implementation |
| Variables declared in an interface are by default final | Abstract class may contain non-final variables as a normal class would contain |
| Java interface is implemented by using keyword "implements" | Abstract class is extended using keyword "extends" |
| Interface can extend another interface only | Abstract class can extend another Java class and implement multiple Java interfaces |
| A Java class can implement multiple interfaces | A Java class can only extend only one abstract class |

In an interview, try not to give answer in a tabular format. It might give a feeling to interviewer that you have mugged up the definitions and differences. Instead try to explain what they are, something like this.

Interface is like a contract with the class, which says I only accept stuff, which looks like the signature that I provide. And class that implements the interface says, sure I will make sure that I look like that.

Interface is kind of a prototype; there are only signatures of methods. Methods do not have any implementation; they do not have behavior of their own.

Abstract class on the other hand as like classes. They look like interface but they can have implementation of methods along with abstract methods. These methods may or may not have behavior and can be overridden in the class that extends the abstract class.

It like saying, abstract class tell the classes extending it, that all the classes should have implemented methods in common (might be overridden) and then these classes should implement the abstract methods also.