The key technical differences between abstract classes and interfaces:

1. Abstract classes can have constants, members, method stubs (methods without a body) and defined methods, whereas interfaces can only have constants and methods stubs.
2. Methods and members of an abstract class can be defined with any visibility, whereas all methods of an interface must be defined as public (they are defined public by default).
3. When inheriting an abstract class, a concrete child class must be define the abstract methods, whereas an abstract class can extend another abstract class and abstract methods from the parent class don’t have to be defined.
4. An interface extending another interface is not responsible for implementing methods from the parent interface. This is because interfaces cannot define any implementation.
5. A child class can only extend a single class (abstract or concrete), whereas an interface can extend or a class can implement multiple other interfaces.
6. A child class can define abstract methods with the same or less restrictive visibility, whereas a class implementing an interface must define the methods with the exact same visibility (public).

Via : stackoverflow