Difference between interface and abstract class

1. First we have to determine when we have to go for interface and when we have to go abstract

Interface: When don’t know the requirements or we don’t know about the implementation but we just have only specifications.

Abstract: When we know about the partial implementation.

1. Method: Public and Abstract

Interface: Every method is public and abstract i.e. pure 100 % abstract.

Abstract class: Both concrete and abstract methods.

1. No final or static methods

Since interface methods can only be abstract then we can’t declare them as final or static.

Why not final: If we declare methods as final, then no sub class can extend them.

Why not static:

But there is not restriction for access modifiers for abstract classes.

1. Interface: Variable as only final, static and public only

At variable level: Every variable in interface is always final, static and public whether we declare it or not.

1. Interface: No transient and synchronized as variables

Since transient and synchronized are applicable with objects only and as also we know no objects can be created for objects so transient and synchronized cannot be applied with interfaces.

1. Interface: No variable as volatile

Because, all variables are final.

1. Interface: Variable initialization compulsory at the time of declaration otherwise compile time error.
2. Interface: No static block declaration
3. Interface: No constructor