

1)Dependency Injection is a design pattern that allows the creation of dependent objects outside of a class and provides those objects to a class through different ways. It focus on moving the creation and binding of the dependent objects outside of the class that depends on them.

2)Dagger is a fully static, compile-time dependency injection framework for both Java and Android. It is an adaptation of an earlier version created by Square and now maintained by Google.

3)Dagger 2 validate graph at Compile-time, while Dagger 1 compose graph at runtime.

Dagger 2 is more easily debugged with concrete stack call for provision and creation

Dagger 2 is fully traceable while Dagger 1 is only partially traceable

Dagger 2 use POJO api while Dagger 1 use map-like api

4)A dependency graph is a directed graph representing dependencies of several objects towards each other.

5)@Module: used on classes which contain methods annotated with provide

@Provides: used for method which provide objects for dependency injection

@Component:Used on an interface. This interface is used by Dagger 2 to generate code which uses the modules to fulfill the requested dependencies.

6)

constructor injection: the dependencies are provided through a class constructor.

setter injection: the client exposes a setter method that the injector uses to inject the dependency.

interface injection: the dependency provides an injector method that will inject the dependency into any client passed to it. Clients must implement an interface that exposes a setter method that accepts the dependency.