**Protocols v/s Subclasses**

**Protocols:-**

\*Protocols define collections of method signatures that the classes conforming to them promise to implement.

\* Once a class has declared that it conforms to a protocol it as the methods that are declared in that class's interface and the rules of inheritance are exactly the same: subclasses inherit the declaration and implementation of the protocol methods but may also choose to override the superclass implementation.

\*Protocols themselves can be extended to produce new protocols consisting of a superset of the methods in the original protocol.

\*Example:-

@protocol <one\_protocol >:<NSProtocol>

@end

**Sub-Classes:-**

\*A subclass inherits methods and instance variables from its superclass.

\*Sub-classing is just a subclass of another class and it inherits behaviour and members of its parent class.

\*For example, you may have a "car" class which has properties that are common to all cars such color, tires, engine, etc. It also has behaviors that are common to all cars such as drive forward, drive backwards, stop, etc.

\*Example:-

@interface parentclass : NSObject

@end

@interface subclass : parentclass

@end

Here subclass inherits from parent class.