|  |  |
| --- | --- |
| **Subclasses** | **Protocols** |
| 1. The subclass has an extended behaviour of its super class | 1. It define a set of behaviour that is expected of an object in a given situation |
| 1. Useful while inheriting the methods and instance variables | 1. Useful when using delegate patterns |
| 1. Subclasses of the same class have a common ancestor | 1. All the classes having the protocol don’t have a common ancestor |
| 1. NSObject is the foundation class | 1. No foundation protocol as such |
| 1. Classes having a super class have a common set of method and property declarations. | 1. Classes having the same protocol share the same method and property declarations without inheriting from a common ancestor. |
| 1. Identity of the class is important . | 1. The identity of the class that implements the protocol is not of interest but what is of interest is whether it has implementations of the methods the protocol declares |
| 1. It is attached to a class | 1. It is not attached to any class definition |
| @interface sub\_class:super\_class  //methods  @end  @implementation sub\_class  //definitions  @end | @protocol prot1 <NSObject> or :<NSProtocol>  //methods  @end  Implementation:  //Must import protocol file.  @interface class\_name: NSObject <prot1>{ }  //methods  @end |