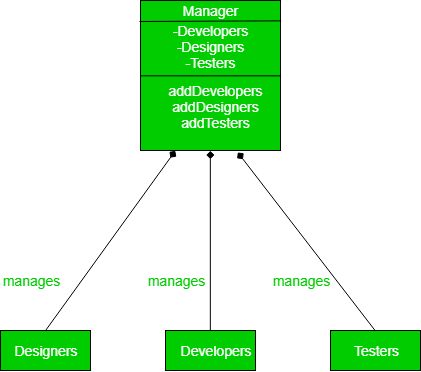
Dependency Inversion Principle (SOLID)

 The main motto of the dependency inversion is *Any higher classes should always depend upon the abstraction of the class rather than the detail*.

 It focuses on the approach where the higher classes are not dependent on the lower classes instead depend upon the abstraction of the lower classes.

**without any dependency inversion**

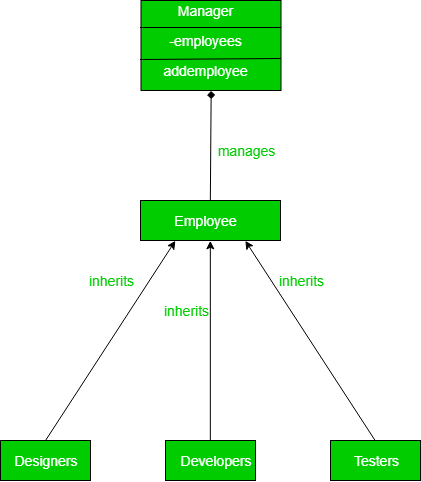


**Drawbacks:-**

We have exposed everything about the lower layer to the upper layer, thus abstraction is not mentioned. That means Manager must already know about the type of the workers that he can supervise.

Now if another type of worker comes under the manager lets say, QA (quality assurance), then the whole class needs to be rejigged. This is where dependency inversion principle pitch in.

**with dependency inversion**



Factory pattern can be example