5. Dependency Inversion =) "High beel module or low lovel moduls in your code should not depend on the actual Emplementation. They should depend on abstraction Interfales Clar A +field: type + method: type Depends on Clars B + field: type + welles : Sope separate on

laro C + fild: type + methode: Sypa of class C K Any change in function which intum will affect class B affect class A. Clan A spield: type + mathed: offer Insorface B class B + fiel + field: Sype + method: sype Ime Shou Induface C dons C + fied : type roll + method: bpe fmethod

& Any changes in methods of class B and C won't affect, Cos the margace remains some. k This is good for writing Jest Coses. * All these SOLID principles ensure the code es Robert, big team modular and allows Change of dreelopers to make Prologendantly