Decorator pattern for weapons used by Ninja:

Decorator pattern is used to add more responsibility to the individual objects and not to all objects. With the decorator pattern each objects will have individual behavior and this behavior can be added at the run time.

In our scenario, our Ninja has various weapon. Each weapon has various strengths and various damages. So, every objects has individual responsibilities and behavior. Now if want to assign various strengths to our weapons, we have to create all combination classes objects for weapon and strengths. So, to eliminate the complexity we are using Decorator pattern for the implementation of weapons.

Classes:

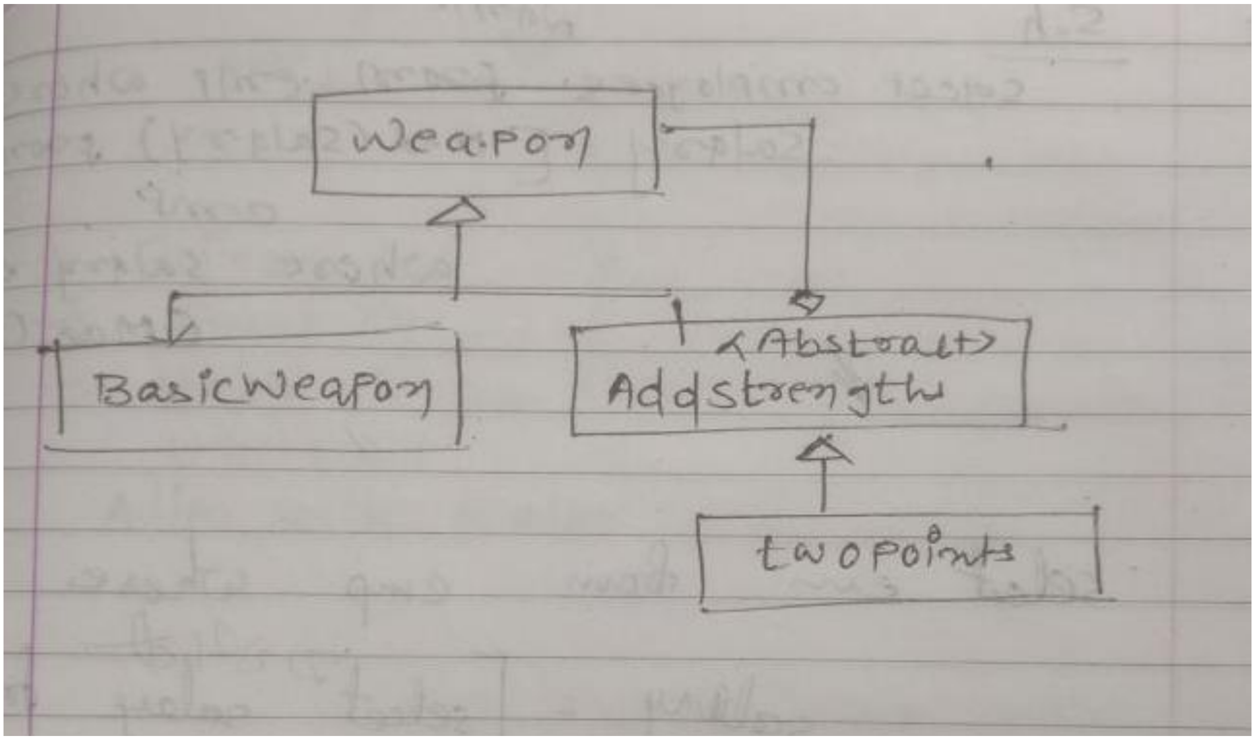
Weapon: Weapon Interface that has to be implemented by all the weapon objects.

BasicWeapon: The most basic weapon which has only minimal damages and powers.

AddStrengths: Abstract class for all the power addition to the Weapon class.

TwoPoints: This is a strength class which can be added to the Weapon class. This class implements AddStrengths class and can have different responsibility then the other children of AddStrengths class.

UML Diagram:


Weapon
BasicWeapon
<Abstract>
Addstrength
twopoints