**SOLID Principles –**

1. **S –** Single Responsibility
2. **O –** Open / Closed
3. **L –** Liskov’s Substitution
4. **I –** Interface Segregation
5. **D –** Dependency Inversion

**\*\*\* Single Responsibility Principals \*\*\***

* Any Function/class/module (unit-of-code) should have a single well-define responsibility
* Every thing should be specify what this class will do, and what this class will not do
* Any piece of code should have only 1 reason to change



**Violating Single Responsibility because in single class should not multiple things**

Class ZooActor{

String name;

String age;

String weight;

void eat();

void poop();

}

Class Animal extends ZooActor{

String species;

String breadName;

Boolean isCarnivorous;

Void roam();

Void lookingVisitor()

}

Class Visitor extends ZooActor {

String address;

String phoneNumber;

Date dateOfBirth;

Void getIdCard();

Void findAnimal();

}

…..

…. More class we can define in a well manner--

**Correcting code after apply Single Responsibility**