Inheritance

Inheritance is a way to form new classes using classes that have already been defined. The newly formed classes are called derived classes, the classes that we derive from are called base classes. Important benefits of inheritance are code reuse and reduction of complexity of a program. The derived classes (child classes) override or extend the functionality of base classes (Parent classes).

Let's see an example by incorporating our previous lecture's work on the Dog class:

In [1]:

```
class Animal:
    def __init__(self):
        print("Animal created")

    def whoAmI(self):
        print("Animal")

    def eat(self):
        print("Eating")

class Dog(Animal):
    def __init__(self):
        Animal.__init__(self)
        print("Dog created")

    def whoAmI(self):
        print("Dog")

    def bark(self):
        print("Woof!")
```

In [2]:

```
d = Dog()
```

```
Animal created
Dog created
```

In [3]:

```
d.eat()
```

```
Eating
```

In [4]:

```
d.whoAmI()
```

```
Dog
```

In [5]:

```
d.bark()
```

```
Woof!
```

In the above example, we have two classes: Animal and Dog. The Animal is the base class, the Dog is the derived class.

The derived class inherits the functionality of the base class.

- **It is shown by the eat() method.**

The derived class modifies existing behavior of the base class.

- **shown by the whoAmI() method.**

Finally, the derived class extends the functionality of the base class, by defining a new bark() method.

In [ ]:

```
```