```
#Inheritance



class Employee:
    company = "ITC"
    def show(self):
        print(f"The name is {self.name} and the salary is {self.salary}")
class programmer:
     company ="ITC Infotech"
     def show(self):
        print(f"The name is {self.name} and the salary is {self.salary}")

     def showLanguage(self):
        print(f"The name is {self.name} and he is good with {self.language} language ")

a = Employee()
b = programmer()

print(a.company, b.company)
```

```
ITC ITC Infotech
```

```
# single inheritance

# Parent class
class Animal:
    def speak(self):
        print("Animal speaks")

# Child class
class Dog(Animal):
    def bark(self):
        print("Dog barks")

# Object of Dog
d = Dog()
d.speak()   # Inherited from Animal
d.bark()    # Own method
```

```
Animal speaks
Dog barks
```

```
#multilevel inheritance
class Animal:
    def eat(self):
        print("Eating")

class Dog(Animal):
    def bark(self):
        print("Barking")

class Puppy(Dog):
    def weep(self):
        print("Weeping")

p = Puppy()
p.eat()
p.bark()
p.weep()
```

```
Eating
Barking
Weeping
```

```
#multiple inheitance
class Father:
    def gardening(self):
        print("I enjoy gardening")

class Mother:
    def cooking(self):
        print("I love cooking")
```

```
class Child(Father, Mother):
    def sports(self):
        print("I play football")

c = Child()
c.gardening()
c.cooking()
c.sports()
```

```
I enjoy gardening
I love cooking
I play football
```

```
#Using super() to access parent class
class Person:
    def __init__(self, name):
        self.name = name

    def show(self):
        print(f"Name: {self.name}")

class Student(Person):
    def __init__(self, name, roll):
        super().__init__(name)
        self.roll = roll

    def show(self):
        super().show()
        print(f"Roll: {self.roll}")

s = Student("Ram", 101)
s.show()
```

```
Name: Ram
Roll: 101
```

Start coding or generate with AI.