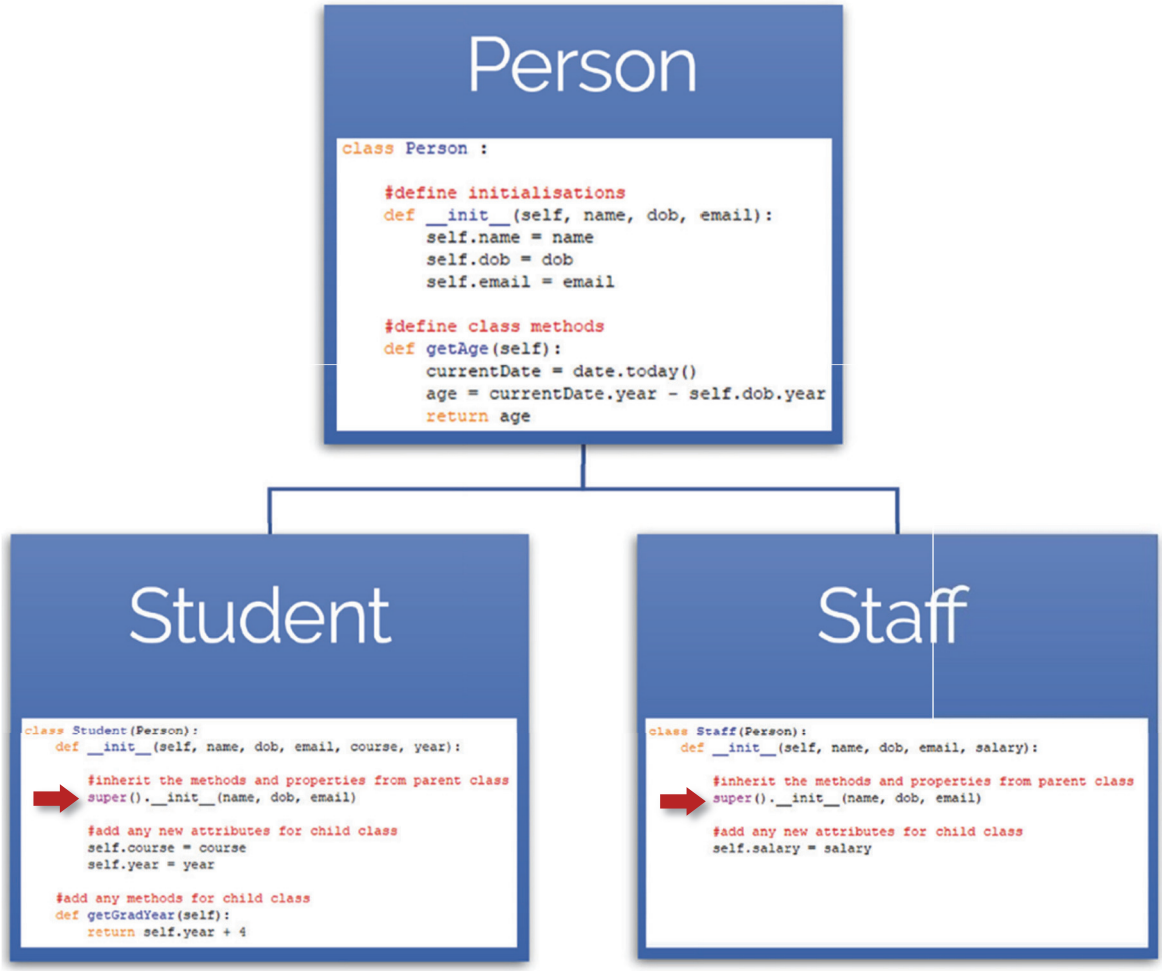
Inheritance

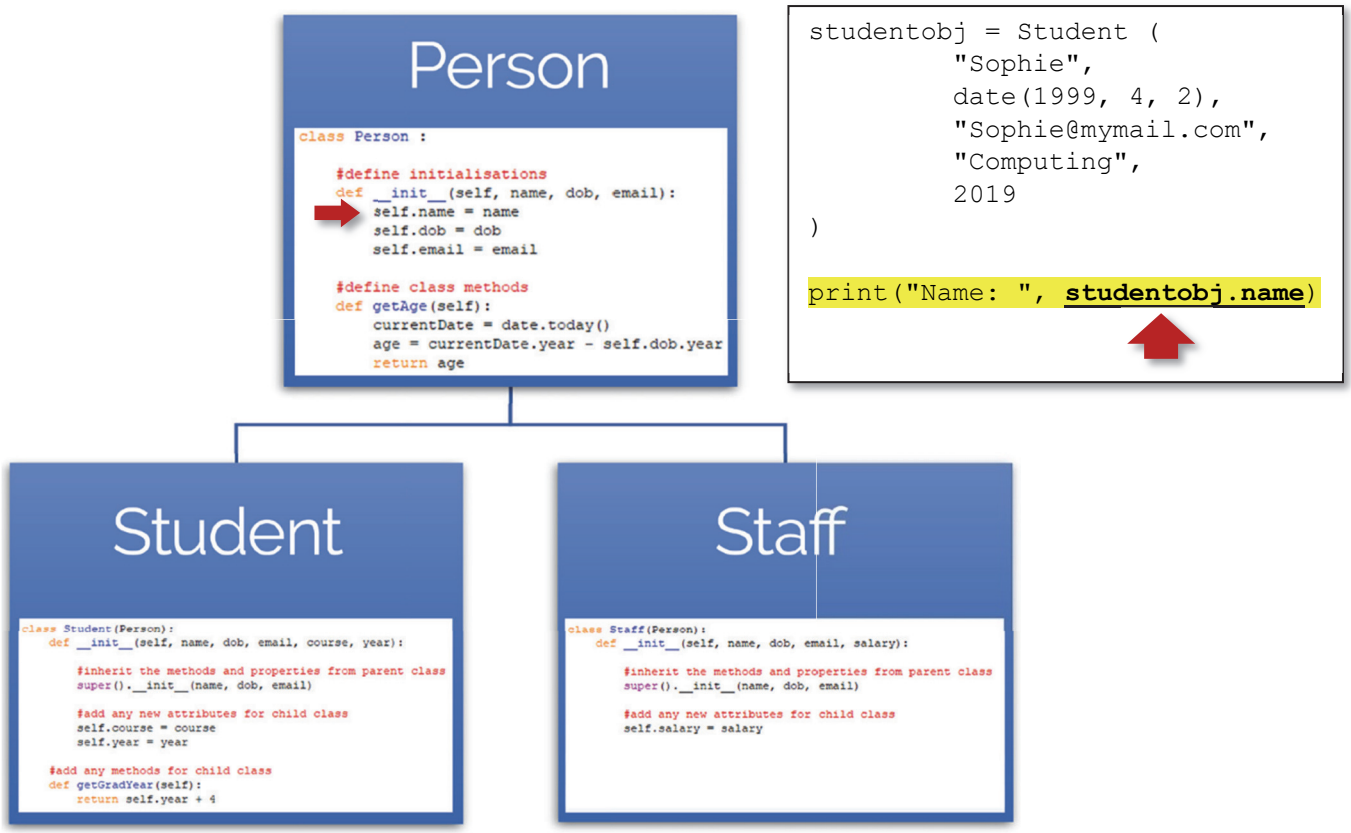
A class can inherit all the methods and attributes from another class. If a class called ‘person’ had name, age, and dob. We could define two other child classes called ‘student’ and ‘staff’. Both can inherit the methods and attributes from the ‘person’ class.



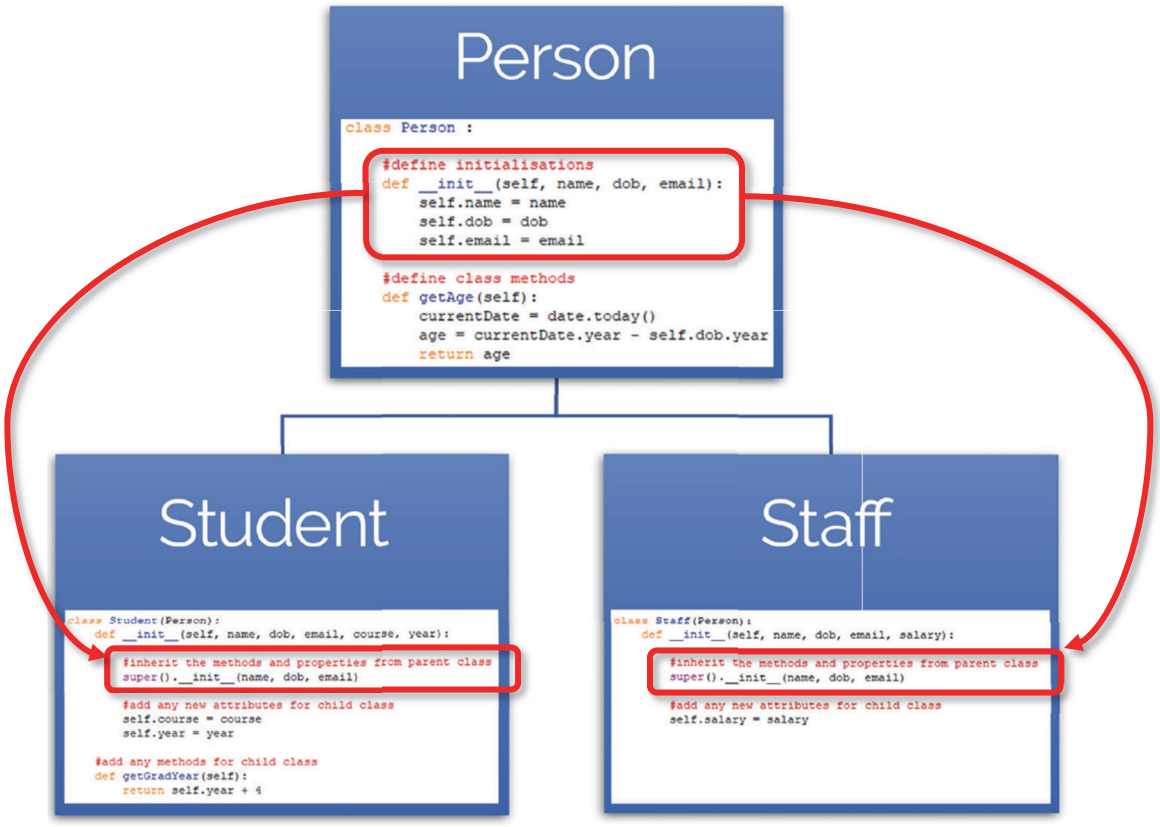
Using the `super()` function in a child class, you can gain access to all the attributes inherited from the parent class and you can call them from an object created using the child class.

We use the `super()` function when we are calling attribute defined in the parent from the child class to make use of it.

In the example below, in the print statement we are calling the “name” attribute, which is defined in the parent class Person, from the object (studentobj) created using the Student class. To access this attribute from a child class, we need to use the `super()` function



To access this attribute defined in the parent from a child class, we need to use the `super()` function in the `__init__()` method of the child class.



The `super()` function calls the the `__init__()` method of parent class, which gives the child class access to all the attributes of the parent class.

If we don't need to access the attributes defined in the parent class from an object created from the child class, then we don't need the `super()` function.

Here we've defined a parent class called Polygon, and created a child class called Triangle.

```
class Polygon:
    def __init__(self, width, height):
        self.width = width
        self.height = height
    def getName(self):
        print ("Polygon")

class Triangle(Polygon):
    def getArea(self):
        return self.width * self.height / 2
```

When we create an object from the child class Triangle…

```
triobj = Triangle(3,4)
```

…we are not calling any attributes defined in the parent class (Polygon). Here, we're calling the `getArea()` method from the triangle object (triobj) defined in the Triangle class.

```
print(triobj.getArea())
```

Similarly if we call the `getName()` method defined in the parent class (Polygon) from a child object (triobj), we don't need `super()`

```
triobj.getName()
```

So when you need to reference an attribute defined in the parent class from an object created by the child class, you need the `super()` function.

Have a look at inherit.py. What happens when you remove the `super()` function from the child class declarations? Why do you need `super()`?