PGR107
Python Programming

Lecture 10 – Inheritance

Kristiania University College

Chapter 10 - Inheritance

Chapter Goals

- To learn about inheritance
- To implement subclasses that inherit and override superclass methods

Inheritance

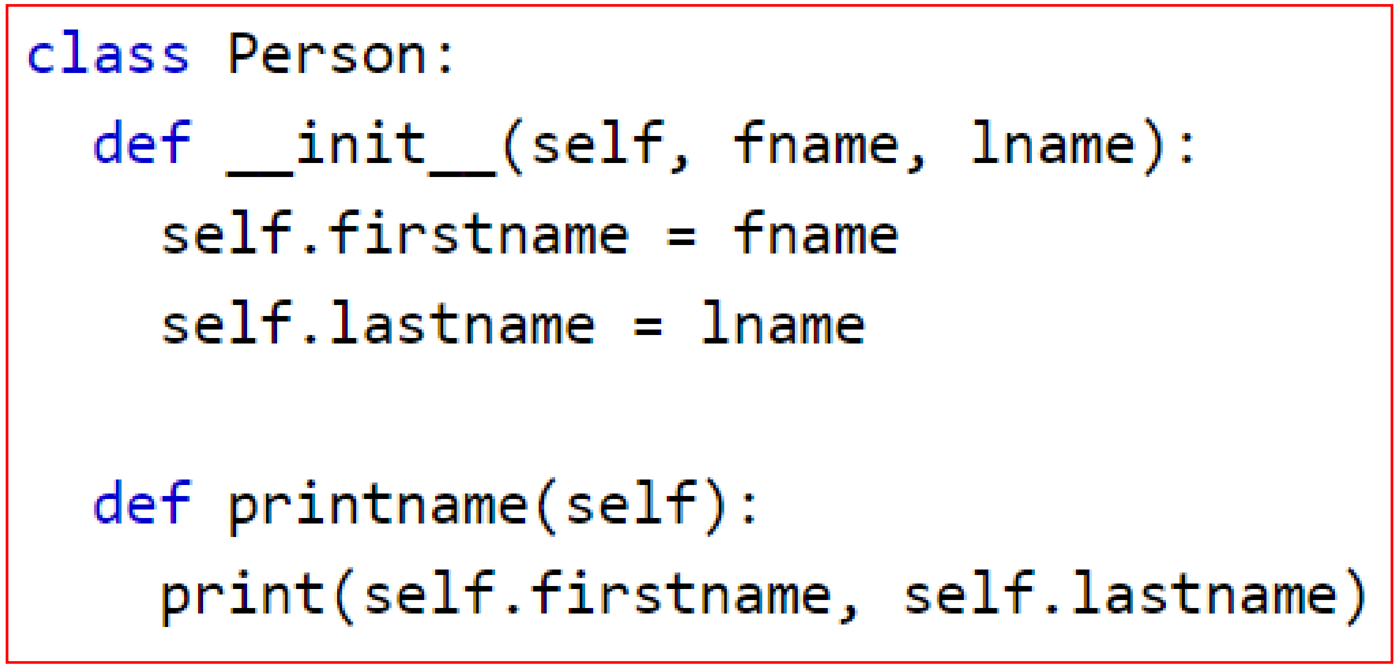
- Inheritance allows you to define a class that inherits all the methods and properties from another class.

- **Parent class** is the class being inherited from, also called base class.

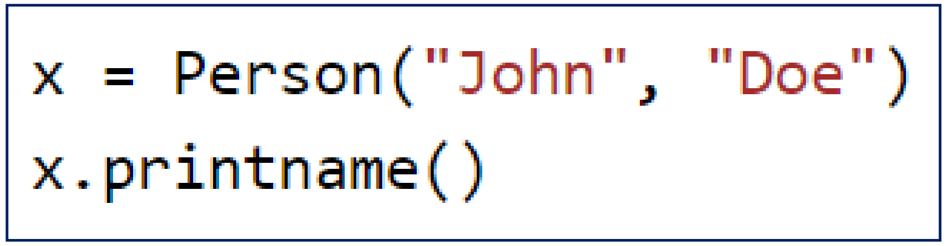
- **Child class** is the class that inherits from another class, also called derived class

Inheritance

- **Parent class:**



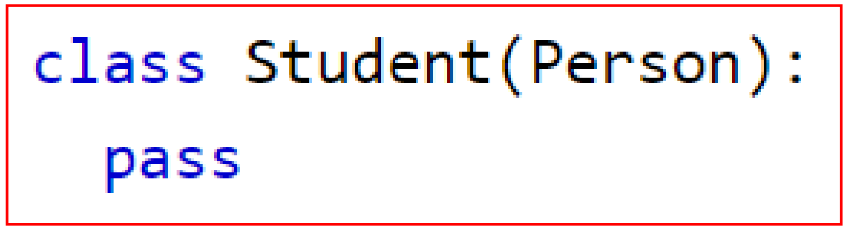
```
class Person:
  def __init__(self, fname, lname):
    self.firstname = fname
    self.lastname = lname

  def printname(self):
    print(self.firstname, self.lastname)
```



```
x = Person("John", "Doe")
x.printname()
```

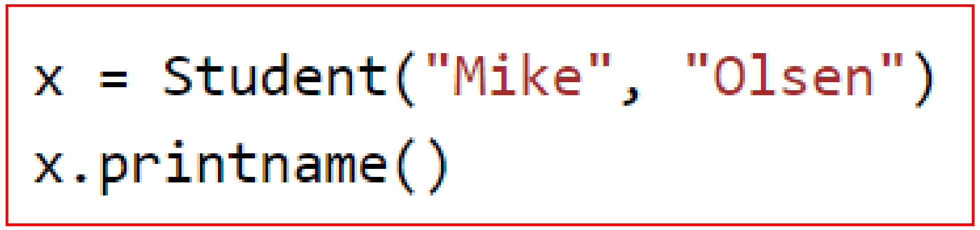
Inheritance

- **Create a Child Class:** To create a class that inherits the functionality from another class, send the parent class as a parameter when creating the child class.



```
class Student(Person):
  pass
```

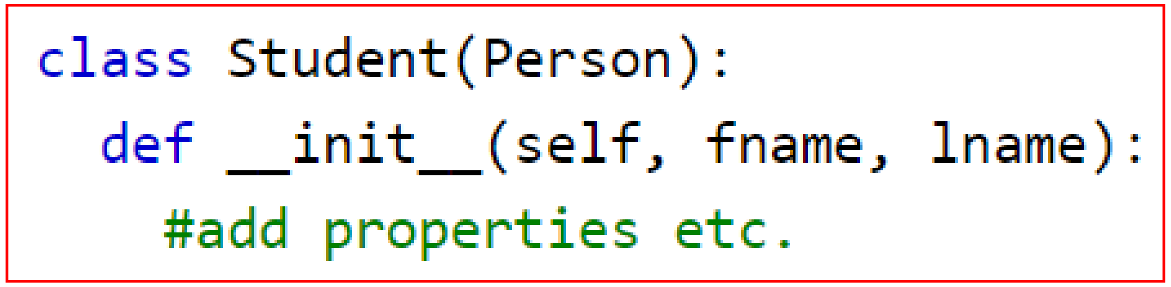
- Use the **pass** keyword when you do not want to add any other properties or methods to the class.



```
x = Student("Mike", "Olsen")
x.printname()
```

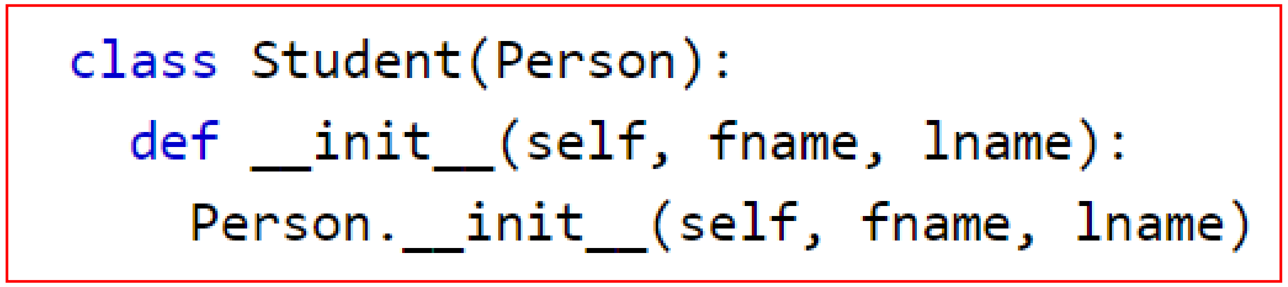
Inheritance

- **Add the __init__() Function:** When you add the **__init__()** function the child class will no longer inherit the parent's **__init__()** function.



```
class Student(Person):
  def __init__(self, fname, lname):
    #add properties etc.
```

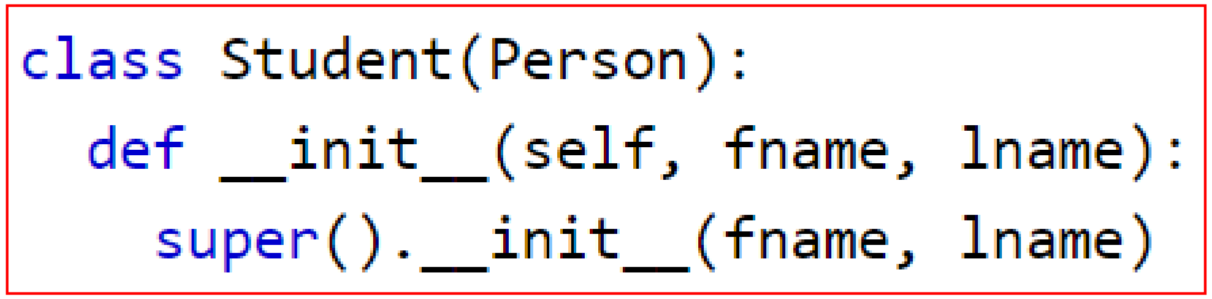
- To keep the inheritance of the parent's **__init__()** function, add a call to the parent's **__init__()** function.



```
class Student(Person):
  def __init__(self, fname, lname):
    Person.__init__(self, fname, lname)
```

Inheritance

- **Use the super() Function:** Python also has a **super()** function that will make the child class inherit all the methods and properties from its parent:

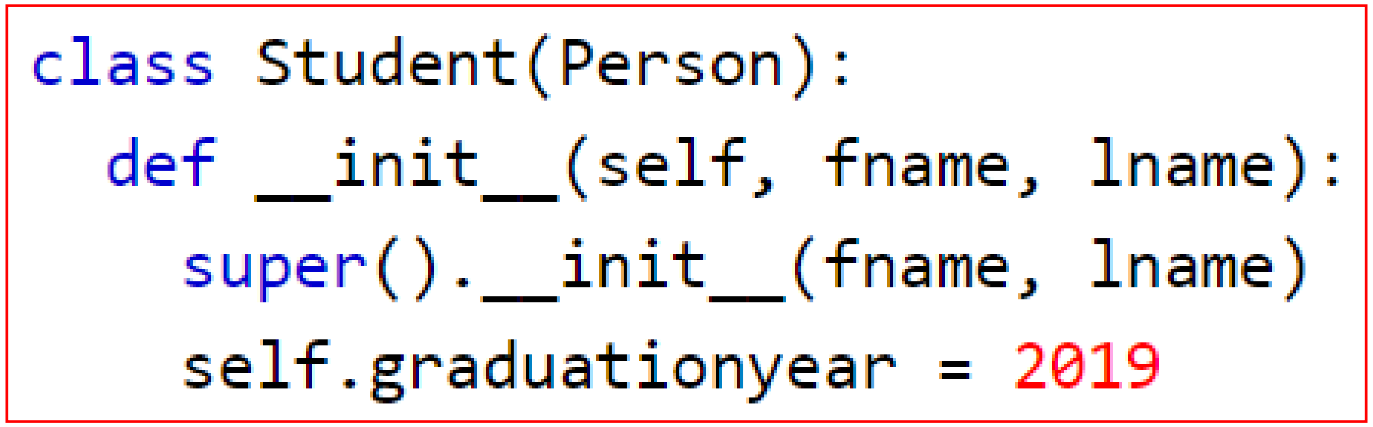


```
class Student(Person):
  def __init__(self, fname, lname):
    super().__init__(fname, lname)
```

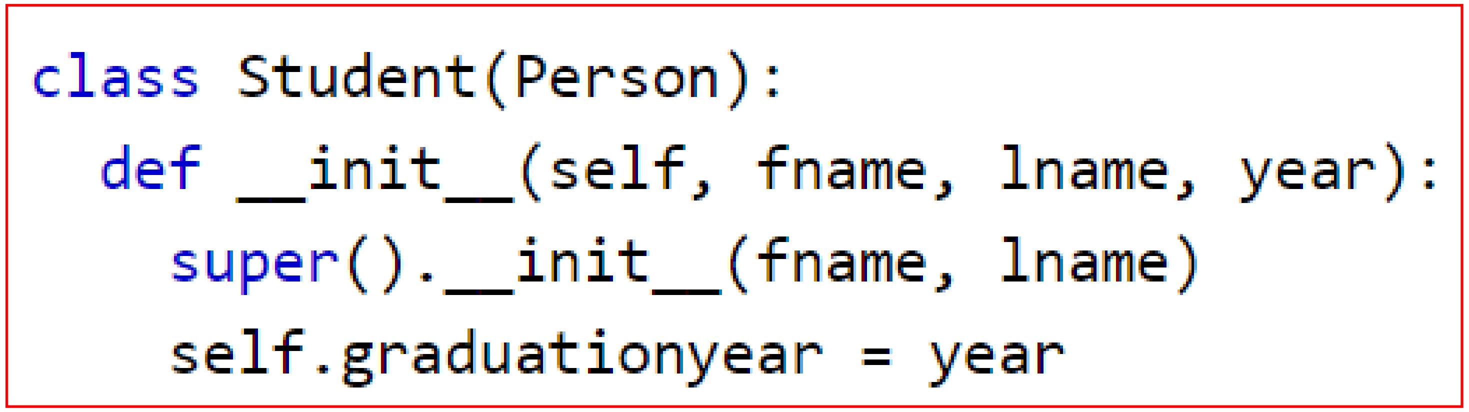
- By using the **super()** function, you do not have to use the name of the parent element, it will automatically inherit the methods and properties from its parent.

Inheritance

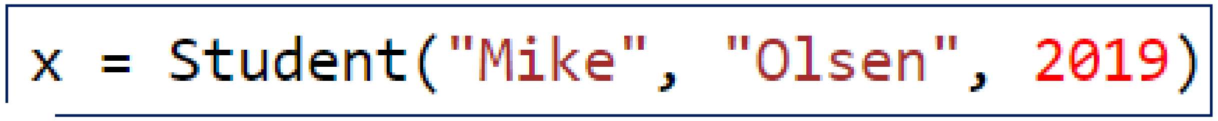
- **Add Properties to Child Class:**



```
class Student(Person):
  def __init__(self, fname, lname):
    super().__init__(fname, lname)
    self.graduationyear = 2019
```



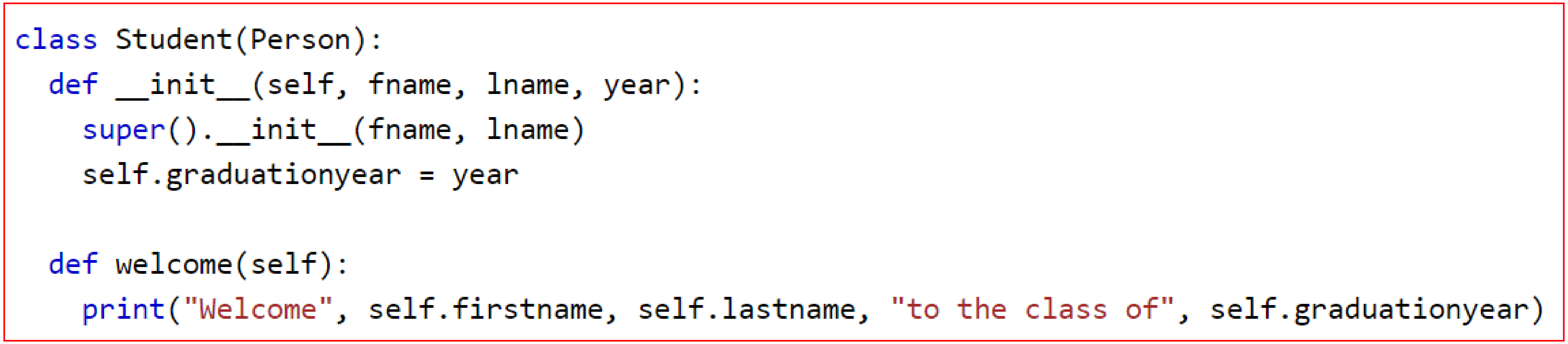
```
class Student(Person):
  def __init__(self, fname, lname, year):
    super().__init__(fname, lname)
    self.graduationyear = year
```



```
x = Student("Mike", "Olsen", 2019)
```

Inheritance

- **Add Methods to Child Class:**



```
class Student(Person):
  def __init__(self, fname, lname, year):
    super().__init__(fname, lname)
    self.graduationyear = year

  def welcome(self):
    print("Welcome", self.firstname, self.lastname, "to the class of", self.graduationyear)
```

- If you add a method in the child class with the same name as a function in the parent class, the inheritance of the parent method will be overridden.

End of Chapter 10