GR5260
Programming for Quantitative & Computational Finance

Instructor: Ka Yi Ng

March 31, 2023

Introduction to OOP

Object-Oriented Programming

- Learnt so far: procedural programming (ie. based on functions and procedures)
- OOP: centered around objects
- Abstraction of concepts from various concrete use cases
- Benefits:
  - More re-usability of code
  - More organized (at least it allows you to do so)
  - Hide complexity from users who call the objects and methods
- Costs:
  - Good object design upfront
  - Changing object design later may require changing common code being used by other use cases

Class definition

- Class instantiation
  - Creating an instance object of a class
- Data variables
  - Instance variables – by instance object
  - Class variables – by class
- Methods:
  - Instance methods – depend on the instance object
  - Static methods – define utility methods independent of the instance/class
  - Class methods – independent of instance object
- Garbage collector:
  - automatic cleanup of memory usage in python

Encapsulation

- Public: accessible to everyone
- Private members: __xxx, only accessible by the class
- Protected: _xxx, only accessible by the class and its subclasses
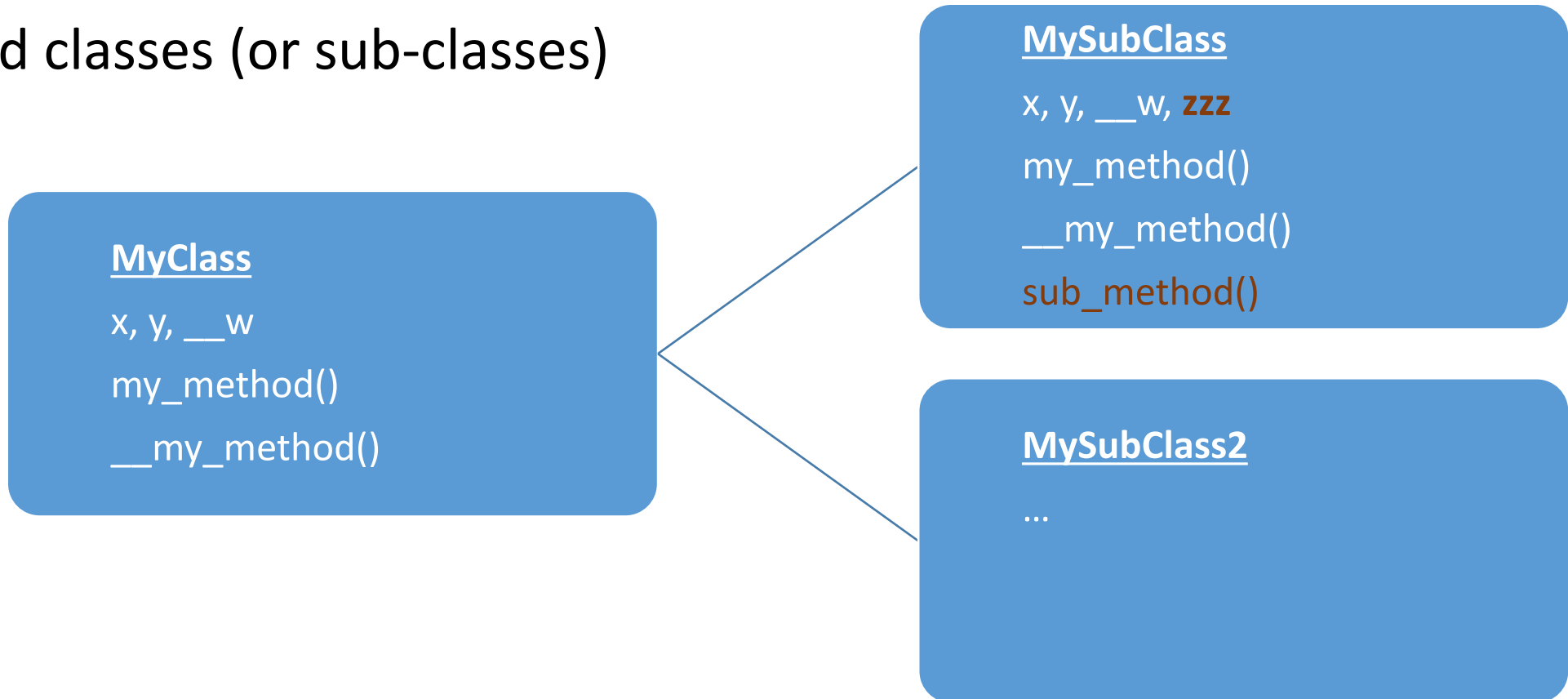- Python does not have native access modifiers to guard the data access
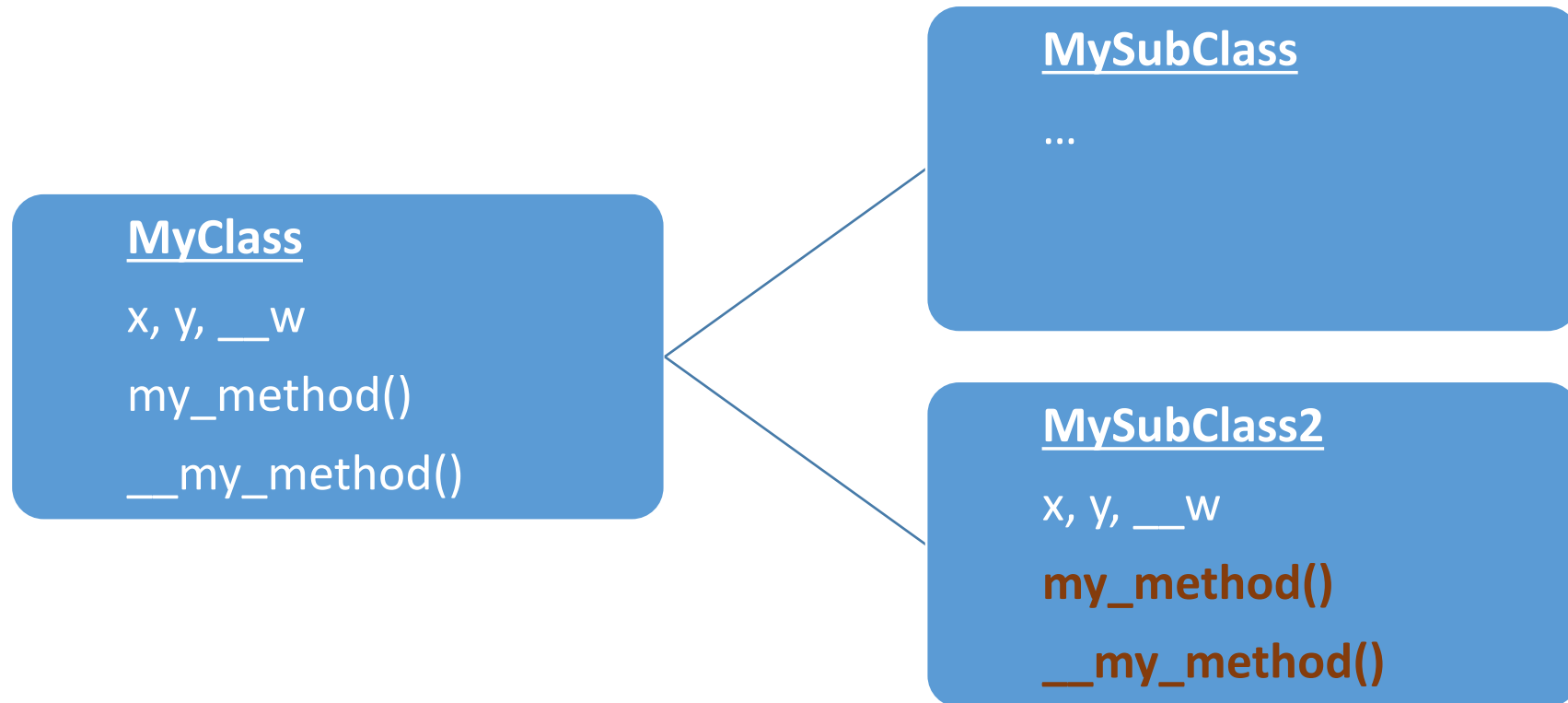
MyClass

x, y

__w

my_method()

__my_method()

Inheritance

- Hierarchy of classes
- Base class
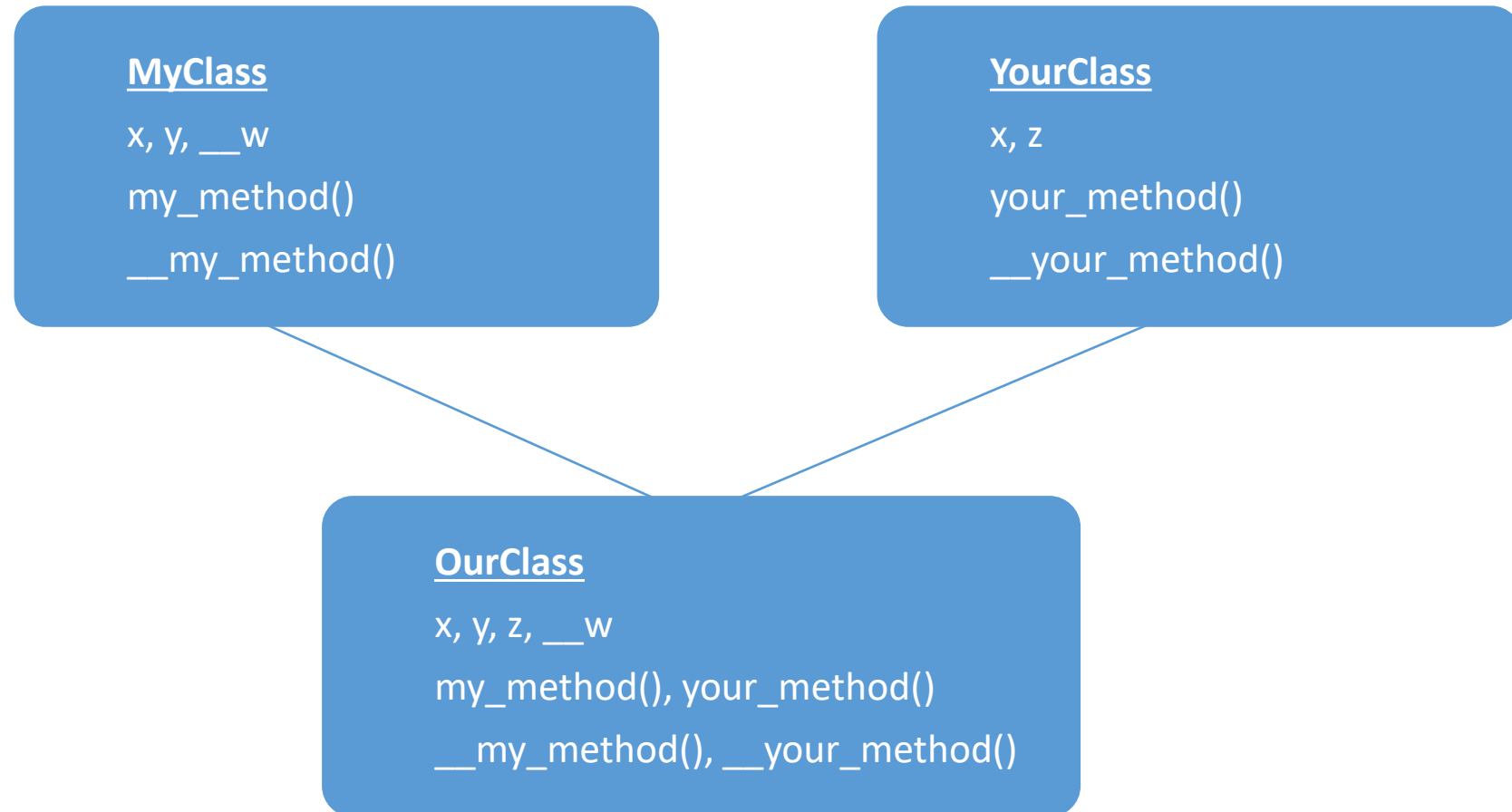- Derived classes (or sub-classes)

MyClass

x, y, __w

my_method()

__my_method()

MySubClass

x, y, __w, **zzz**

my_method()

__my_method()

sub_method()

MySubClass2

…

Polymorphism

- Different implementation of a base class method in different sub-classes

MySubClass

...

MyClass

x, y, __w

my_method()

__my_method()

MySubClass2

x, y, __w

my_method()

__my_method()

Multiple inheritance

- Derived from multiple base classes

MyClass

x, y, __w

my_method()

__my_method()

YourClass

x, z

your_method()

__your_method()

OurClass

x, y, z, __w

my_method(), your_method()

__my_method(), __your_method()

Composition

- Consists of multiple classes

MyClass

x, y, __w

my_method()

__my_method()

YourClass

x, z

your_method()

__your_method()

OurClass

An instance of MyClass

An instance of OurClass
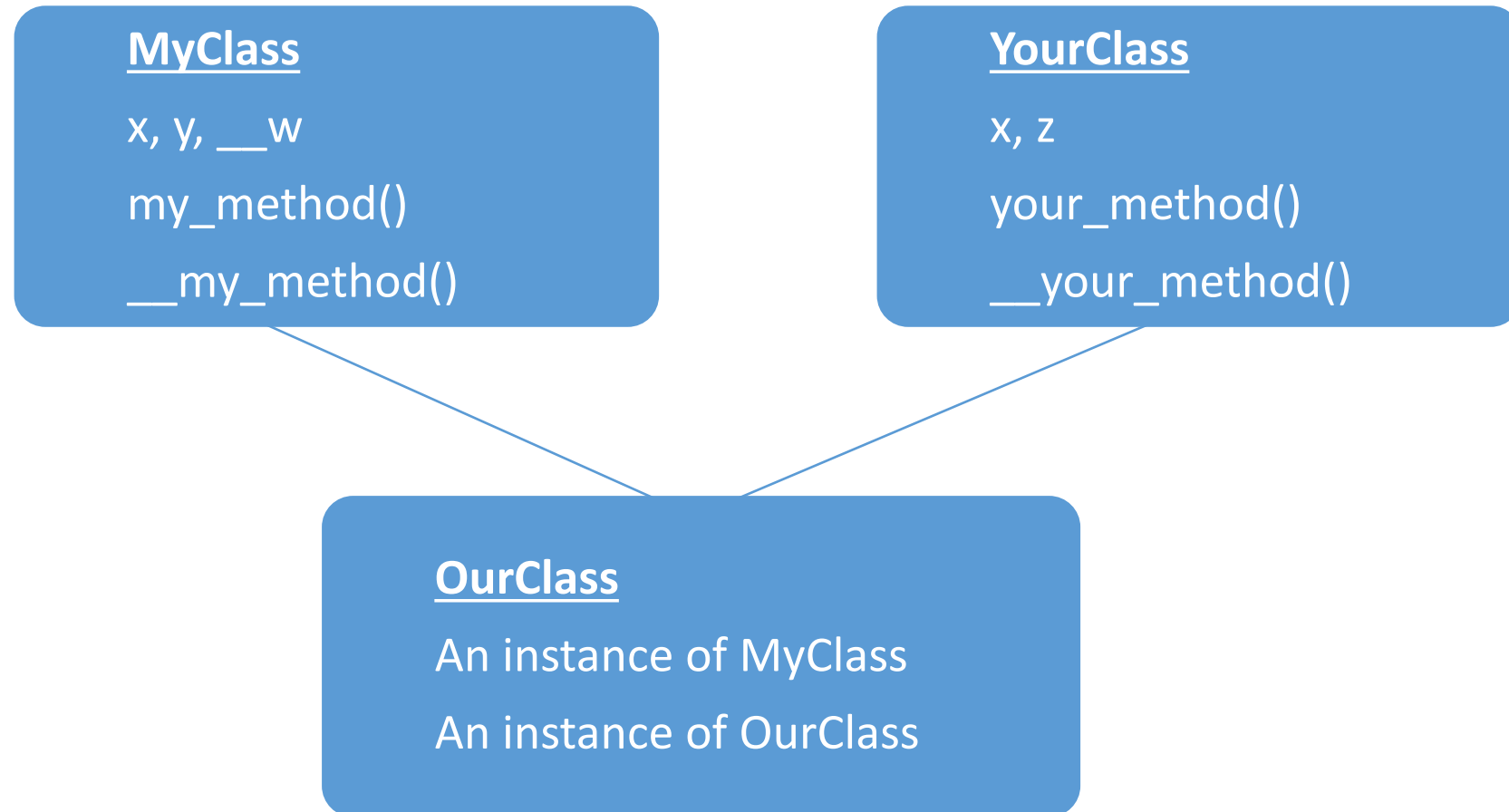
Example: Trades on equities

- Trade: side, quantity, trade date, settle date, etc.
  - Vanilla: equity name
  - EuropeanOption: equity, strike, option expiry, call/put
  - AsianOption: equity, strike, option expiry, average start, average end, call/put
  - SpreadOption: equity1, equity2, strike, option expiry, call/put
  - BasketOption: equity_list, strike, option expiry, call/put
  - etc