# Inheritance

Object Oriented programming

Abstract datatype

Public interface

Private internal implementation

Encapsulation

Reusing datatypes

Hierarchy of classes

Class 1

|

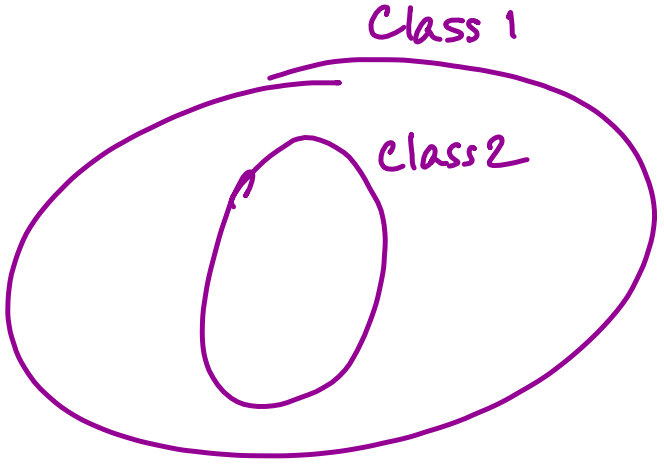
Class 2

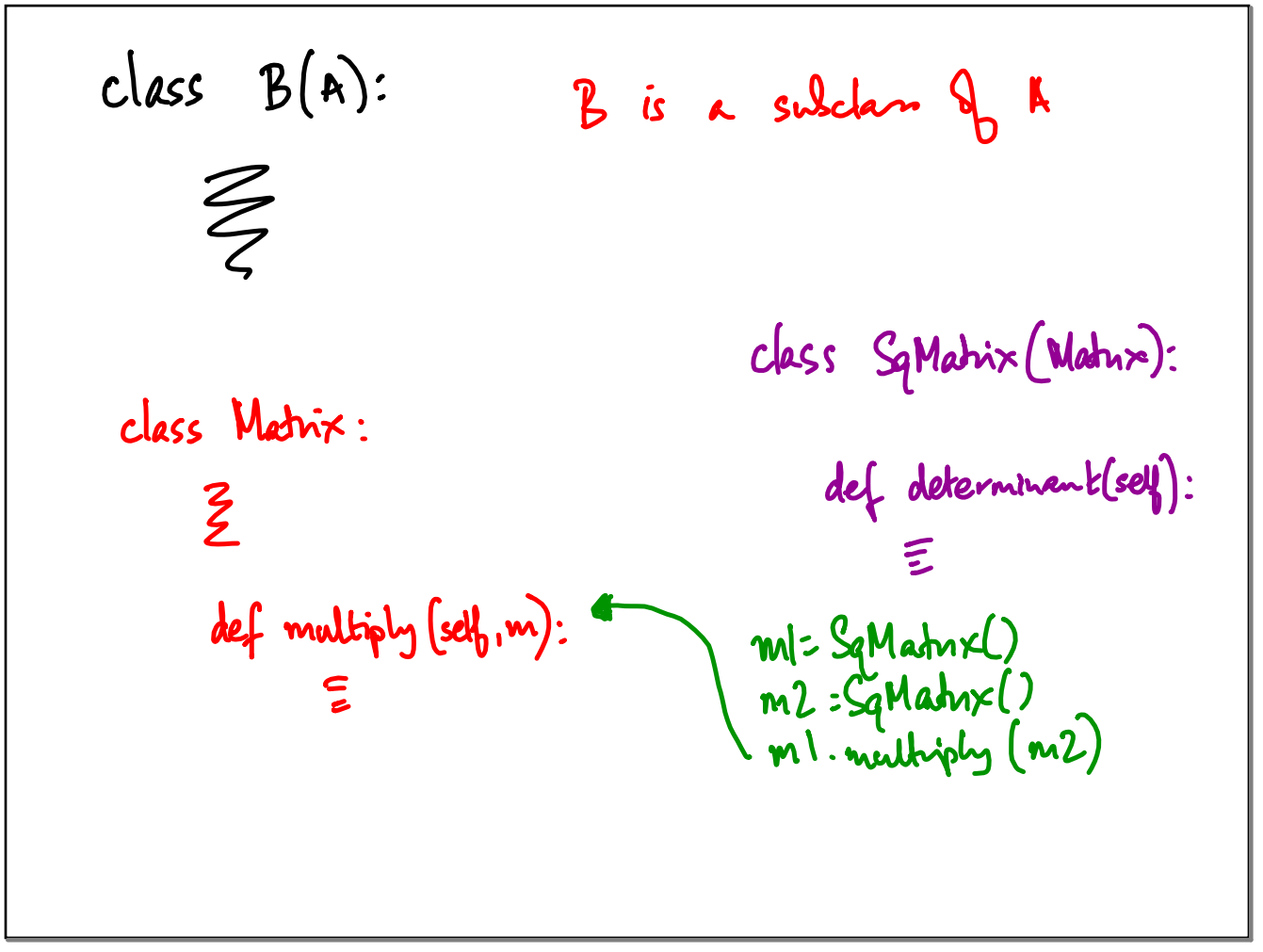
Class 1 — Matrices
superclass

Class 2 can do everything Class 1 does, and more

Class 2 subclass

Square Matrices




class B(A):
B is a subclass of A
class SqMatrix(Matrix):
class Matrix:
def determinant(self):
def multiply(self,m):
m1 = SqMatrix()
m2 = SqMatrix()
m1.multiply(m2)

Further complications

(1) Parent has a function f()
Child wants a different version of f()

Redefine f() in Child
Automatically choose nearest one

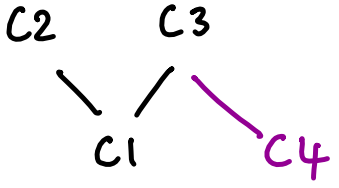
super refers to parent class
self refers to this obj

(2) C1 — f()
|
C2(C1) — f()
|
C3(C2) — Can I ask for C1.f()?

super.f()
super.super.f()

## Can I inherit from more than one class?

```
class C1(C2,C3):
```



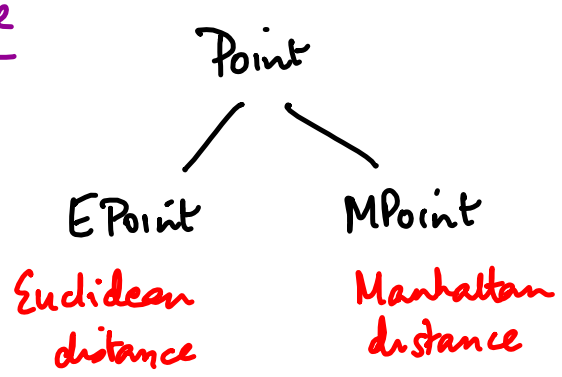

What if f() is defined in both C2, C3 but not in C1

Resolve by searching in sequence: C2 then C3

```
super(C3, self).f()
```

self is treated as instance of C3, look for f() in C3

Example


Point
EPoint
MPoint
Euclidean distance
Manhattan distance