[Self-review: Working with Methods | Coursera](https://www.coursera.org/learn/programming-in-python/quiz/71Mfg/self-review-working-with-methods/attempt?redirectToCover=true)

**True or False: A class can serve as a base class for many derived classes.**

(C) True

Correct! Theoretically, a class can be a superclass for an infinite number of subclasses, allowing for extensive reuse of base class functionality.

(X) False

**In case of multiple inheritance where C is a derived class inheriting from both class A and B, and where a and b are the respective objects for these classes, which of the following code will inherit the classes A and B correctly? (Select all that apply)**

(C) class C(B, A)

*Correct! Python supports multiple inheritance, allowing class C to inherit from both classes A and B, regardless of the order they are listed.*

(C) class C(A, B)

*Correct! The order of parent classes in the declaration doesn’t affect the ability of the derived class to inherit from both.*

(X) class(a, B)

*Incorrect. This syntax is incorrect for class definition; class names should be capitalized, and the declaration does not use parentheses around class names.*

(X) class (a, b)

*Incorrect. This syntax is incorrect for defining a class. Class names must be capitalized, and this does not correctly define inheritance in Python.*

**In Example 3, if we modified the code to include a global variable ‘a = 5’ as follows:**

**Python**

a = 5

class A:

a = 7

pass

class B(A):

pass

class C(B):

pass

Will the code work and what will be the output if it does?

(C) Yes and it will print the value 7

Correct! Classes B and C inherit from class A, where 'a' is defined as 7. Accessing 'a' through these classes will refer to this definition, not the global 'a'.

(X) No

(X) Yes and it will print the value 5

Incorrect. While a global variable 'a' is defined, the class attribute 'a' in class A shadows the global variable within the context of the class and its subclasses.

**What function can be used other than mro() to see the way classes are inherited in a given piece of code?**

(X) info()

(X) dir()

(X) class()

(C) help()

*Correct! The help() function can be used to see detailed information about classes, including the inheritance chain, which complements the information provided by the mro() method.*

**The super() function is \_\_\_\_? (Select all that apply)**

(C) called inside the child class init()

*Correct! It's commonly used in the child class's \_\_init\_\_() to ensure that the parent class's \_\_init\_\_() method is called, facilitating proper initialization.*

(C) used to call different parent class method

*Correct. While not specified as correct in the original formatting, super() can indeed be used to call methods from the parent class other than \_\_init\_\_().*

(X) called over the init() method of the class it is called from

*Incorrect. super() is used to call methods from the parent class, not from the class it is called within.*

What is the type of inheritance in the code below:

python

class A():

pass

class B(A):

pass

class C(B):

pass

(X) Multiple

(X) Hierarchical

(C) Multi-level

*Correct! This pattern of inheritance, where a class inherits from a class that is also a derived class, is known as multi-level inheritance.*

(X) Single

*Incorrect. While single inheritance refers to a class deriving from one parent class, this scenario involves multiple levels of inheritance.*