Congratulations! You passed!

Grade received 100% **To pass** 80% or higher

Go to next item

1. What is the purpose of the self parameter in Python class methods? **1 / 1 point**

- ○ To allow methods to modify the class state
- ○ To reference the parent class
- ○ To automatically pass arguments
- ◉ To access the instance's attributes

Correct
Correct, self refers to the current instance.

2. How does a Python subclass inherit behavior from a parent class? **1 / 1 point**

- ○ The subclass overrides all parent behavior
- ◉ The subclass class definition lists the parent
- ○ The parent constructor configures subclass state
- ○ Inheritance happens automatically without configuration

Correct
Correct, inheritance happens by subclass listing parent by name.

3. What is a benefit of using a constructor in a Python class? **1 / 1 point**

- ◉ It sets initial values and state
- ○ It defines default method implementations
- ○ It automatically inherits from object
- ○ Nothing, constructors provide no benefit

Correct
Correct, constructors initialize new instances with default values.

4. When would you use class inheritance in your code? **1 / 1 point**

- ○ To reuse only method implementations
- ◉ To share attributes and behaviors between classes
- ○ To make subclasses depend on parent state
- ○ Only when working with existing inherited code

Correct
Correct, inheritance promotes reuse of common logic across classes.

5. Why use the self prefix inside Python class methods? **1 / 1 point**

- ○ It refers to the global scope
- ○ It is required by Python's syntax
- ○ It creates new local variables
- ◉ It maps names to instance attributes

Correct
Correct, self.name stores values on the instance.