Polymorphism and Abstraction

Total points 9/9

Theory test about

Polymorphism and Abstraction in Python.

✓ What is polymorphism in Python? * 1/1

- (●) The ability of an object to take on many forms ✓
- ( ) The process of converting data types
- ( ) The ability of an object to inherit from multiple classes
- ( ) The process of breaking down a complex problem into smaller sub-problems

✓ What is abstraction in Python? * 1/1

- ( ) A technique for controlling access to class members
- (●) The process of hiding the implementation details of a class ✓
- ( ) A technique for reducing the size of an object
- ( ) A process of adding new features to a class

✓ What is overloading in Python? * 1/1

- ○ The process of assigning multiple values to a variable
- ○ The process of creating multiple instances of a class
- ◉ The process of changing the behavior of functions by creating new functions with the same name.. ✓
- ○ The process of hiding the implementation details of a class

✓ What is duck typing in Python? * 1/1

- ○ The process of determining an object's type at runtime
- ○ The process of explicitly specifying the type of an object
- ○ The process of checking the type of an object before using it
- ◉ The process of checking for the presence of a given method or attribute no matter the object's type. ✓

✓ Which module is used to define abstract classes in Python? * 1/1

- ○ os module
- ◉ abc module ✓
- ○ sys module
- ○ math module

✓ What is the purpose of @abstractmethod decorator in Python? * 1/1

- ○ To mark a method as a static method
- ○ To mark a method as a class method
- ◉ To mark a method as an abstract method ✓
- ○ To mark a method as a private method

✓ What method is used to overload "+"? * 1/1

- ○ __plus__
- ◉ __add__ ✓
- ○ __sum__
- ○ __increment__

✓ What is the purpose of abstract classes in Python? * 1/1

- ◉ To provide a blueprint for other classes to follow ✓
- ○ To prevent instances of the class from being created
- ○ To provide a mechanism for multiple inheritance
- ○ To provide a way to define methods with default implementations

✓ What is the purpose of the isinstance() function in Python? * 1/1

- ○ To check if a method has been overridden in a subclass
- ○ To check if an object is of a particular type
- ◉ To check if an object is an instance of a particular class ✓
- ○ To check if a method has been overridden in a class