Vendor: Python Institute

Exam Code: PCPP-32-101

Exam Name: Certified Professional in Python Programming 1

Version: DEMO

QUESTION 1
Select the true statement about composition.

A. Composition extends a class's capabilities by adding new components and modifying the existing ones
B. Composition allows a class to be projected as a container of different classes
C. Composition is a concept that promotes code reusability, while Inheritance promotes encapsulation
D. Composition is based on the has a relation, so it cannot be used together with inheritance

Answer: B
Explanation:
Composition in POO allows a class to act as a container that holds instances of other classes, fostering code reusability and enabling the creation of complex structures by combining simpler components. It establishes a "has-a" relationship, promoting encapsulation and flexibility, as the container class can access the public interface of the contained classes, but not their internal details, facilitating modular and adaptable code design.

QUESTION 2
Analyze the following snippet and select the statement that best describes it.

```
class OwnMath:
    pass

def calculate_value(numerator, denominatior):
    try:
        value = numerator / denominator
    except ZeroDivisionError as e:
        raise OwnMath from e
    return value

calculate_value(4, 0)
```

A. The code is an example of implicitly chained exceptions.
B. The code is erroneous as the OwnMath class does not inherit from any Exception type class
C. The code is fine and the script execution is not interrupted by any exception.
D. The code is an example of explicitly chained exceptions

Answer: B
Explanation:
OwnMath should be derived from BaseException. Run this code and you'll get an exception. If it is derived, it's an example of explicitly chained exception.

QUESTION 3
Analyze the following snippet and select the statement that best describes it.

```
class Sword:
    var1 = 'weapon'

    def __init__(self):
        self.name = 'Excalibur'
```

A. self.name is the name of a class variable
B. var1 is the name of a global variable
C. Excalibur is the value passed to an instance variable
D. weapon is the value passed to an instance variable

Answer: C
Explanation:
Excalibur is the value passed to an instance variable. In the given code snippet, self.name is an instance variable of the Sword class. When an instance of the Sword class is created with varl = Sword('Excalibur'), the value 'Excalibur' is passed as an argument to the __init__ method and assigned to the name instance variable of the varl object. The code defines a class called Sword with an __init__ method that takes one parameter name. When a new instance of the Sword class is created with varl = Sword('Excalibur'), the value of the 'Excalibur' string is passed as an argument to the __init__ method, and assigned to the self.name instance variable of the varl object.

QUESTION 4
The following snippet represents one of the OOP pillars. Which one is that?

```
class A:
    def run(self):
        print("A is running")

class B:
    def fly(self):
        print("B is flying")

class C:
    def run(self):
        print("C is running")

for element in A(), B(), C()
    element.run()
```

A. Serialization
B. Inheritance
C. Encapsulation
D. Polymorphism

Answer: D
Explanation:

The code snippet you provided demonstrates **Polymorphism**. This is because the 'run()' method is called on different objects (instances of classes A, B, and C), and each class defines its own version of the 'run()' method. This allows the same method call to behave differently based on the object it is called on.

QUESTION 5
Analyze the following function and choose the statement that best describes it.

```
def my_decorator(coating):
    def level1_wrapper(my_function):
        def level2_wrapper(*args)
            our_function(*args)
        return level2_wrapper

    return level1_wrapper
```

A. It is an example of a decorator that accepts its own arguments.
B. It is an example of decorator stacking.
C. It is an example of a decorator that can trigger an infinite recursion
D. The function is erroneous.

Answer: D
Explanation:
If you run this code to decorator a new function, you will get name error, regardless the colon in line 3 is missing or not.
NameError: name 'level2_warpper' is not defined.
This is because our_function (line 4) and my_function(line 2) did not return the SAME function name.
line 2: def level1_wrapper(my_function)
line 4: return our_function
To make this code correct, you can make FUNCTION_NAME in line 2 and line 4 consistency. eg. line 4 should be 'return my_function'.

QUESTION 6
Analyze the following snippet and select the statement that best describes it.

```
def f1(*arg, **args):
    pass
```

A. The code is syntactically correct despite the fact that the names of the function parameters do not follow the naming convention
B. The *arg parameter holds a list of unnamed parameters.
C. The code is missing a placeholder for unnamed parameters.
D. The code is syntactically incorrect - the function should be defined as def f1 (*args, **kwargs):

Answer: D
Explanation:
The use of a single asterisk (*) before the parameter name (arg) indicates that it can accept a variable number of unnamed positional arguments. When the function is called, these arguments

will be collected into a tuple named arg.
The **args parameter, with two asterisks (**), allows for a variable number of keyword arguments to be passed to the function. These arguments will be collected into a dictionary named args.

QUESTION 7
Analyze the following snippet and decide whether the code is correct and/or which method should be distinguished as a class method.

```
class Crossword:
    number_of_Crosswords = 0

    def __init__(self, height, width):
        self.height = height
        self.width = width
        self.progress = 0

    @staticmethod
    def isElementCorrect(word):
        if self.isSolved():
            print('The crossword is already solved')
            return True
        result = True
        for char in word:
            if char.isdigit():
                result = False
                break
        return result

    def isSolved(self):
        if self.progress == 100
            return True

    def getNumberOfCrosswords(cls):
        return cls.number_of_Crosswords
```

A. There is only one initializer, so there is no need for a class method
B. The getNumberOfCrosswords() method Should be decorated with @classmethod
C. The code is erroneous
D. The getNumberOfCrosswords() and isSolved methods should be decorated with @classmethod

Answer: C
Explanation:
The isElementCorrect method is decorated as a @staticmethod but incorrectly tries to call self.isSolved(), which is an instance method. Static methods do not have access to self because they do not require an instance of the class to be called. They work at the class level, not the instance level.

QUESTION 8
Analyze the code and choose the best statement that describes it.

```
class Item:
    def __init__(self, initial_value)
        self.value = initial_value

    def __ne__(self, other):
        ...
```

A. __ne__() is not a built-in special method.
B. The code is erroneous
C. The code is responsible for the support of the negation operator, e.g. a = - a
D. The code is responsible for the support of the inequality operator, i.e. !=

Answer: D
Explanation:
In the given code snippet, the __ne__ method is a special method that overrides the behavior of the inequality operator != for instances of the MyClass class. When the inequality operator is used to compare two instances of MyClass, the __ne__ method is called to determine whether the two instances are unequal.

QUESTION 9
Which function or operator should you use to obtain the answer True or False to the question "Do two variables refer to the same object?"

A. The == operator
B. The isinstance() function
C. The id() function
D. The is function

Answer: D
Explanation:
To determine if two variables refer to the same object in Python, you should use the 'is' operator. The 'is' operator checks for object identity, meaning it returns 'True' if both variables point to the same object in memory.

Thank You for Trying Our Product

Passleader Certification Exam Features:

★ More than **99,900** Satisfied Customers Worldwide.

★ Average **99.9%** Success Rate.

★ **Free Update** to match latest and real exam scenarios.

★ **Instant Download** Access! No Setup required.

★ Questions & Answers are downloadable in **PDF** format and **VCE** test engine format.

★ Multi-Platform capabilities - **Windows, Laptop, Mac, Android, iPhone, iPod, iPad**.

★ **100%** Guaranteed Success or **100%** Money Back Guarantee.

★ **Fast**, helpful support **24x7**.

View list of all certification exams: http://www.passleader.com/all-products.html

10% Discount Coupon Code: **ASTR14**