

[Back to Previous Page](#)

Quiz Result For "Python OOP Quiz" Is Here -

🔍 Total Question - 27

📋 Total Attempts - 26

✅ Total Correct - 25

❌ Total Incorrect - 1

🕒 Duration - 27Minutes

📊 Passing Percentage - 70%

📊 Your Percentage - 92.59%

Q1. What is the primary purpose of a class in Python?

- 1) To store data
- 2) To define functions
- 3) To provide a blueprint for creating objects
- 4) To execute code

Your Answer: 3) ✓

To provide a blueprint for creating objects

Correct Answer: 3)

To provide a blueprint for creating objects

Q2. What is the correct way to create an instance of the Dog class?

1)

```
my_dog = Dog.create("Buddy")
```

2)

```
my_dog = Dog("Buddy")
```

3)

```
my_dog = new Dog("Buddy")
```

4)

```
my_dog = create_instance(Dog, "Buddy")
```

Your Answer: 2) ✓

```
my_dog = Dog("Buddy")
```

Correct Answer: 2)

```
my_dog = Dog("Buddy")
```

Q3. What is a class attribute in Python?

- 1) An attribute that is only accessible within the class
- 2) An attribute shared by all instances of a class
- 3) An attribute that can be modified after object creation
- 4) An attribute defined inside a method

Your Answer: 2) ✓

An attribute shared by all instances of a class

Correct Answer: 2)

An attribute shared by all instances of a class

Q4. Here is a class `Animal` with a method `make_sound`. The subclass `Dog` inherits from `Animal`. Choose the best option to override the `make_sound` method to print "Woof!"

1)

```
def bark(self):  
    print("Bark!")
```

2)

```
def sound(self):  
    print("Woof!")
```

3)

```
def make_sound(self):  
    print("Meow!")
```

4)

```
def make_sound(self):  
    print("Woof!")
```

Your Answer: 4) ✓

```
def make_sound(self):  
    print("Woof!")
```

Correct Answer: 4)

```
def make_sound(self):  
    print("Woof!")
```

Q5. Here is a class `Person` with a class attribute `total_people` initialized to 0. In the `__init__` method, we want to increment the `total_people` class attribute, and set the attribute `name`. Choose the best option

1)

```
total_people += 1;  
self.name = name
```

2)

```
Person.total_people += 1;  
self.name = name
```

3)

```
self.total_people += 1;  
self.name = name
```

4)

```
self.total_people = 0;  
self.name = name
```

Your Answer: 2) ✓

```
Person.total_people += 1;  
self.name = name
```

Correct Answer: 2)

```
Person.total_people += 1;  
self.name = name
```

Q6. How do you access the attribute `released_date` of an instance `pokemon` of class `Game` in Python?

1)

```
pokemon.released_date
```

2)

```
MyClass.released_date
```

3)

```
pokemon.get("released_date")
```

4)

```
MyClass.get_attribute(pokemon)
```

Your Answer: 1) ✓

```
pokemon.released_date
```

Correct Answer: 1)

```
pokemon.released_date
```

Q7. How would you set a new balance of 100 to an instance `acc` of BankAccount?

1)

```
acc.__balance = 100
```

2)

```
acc.balance = 100
```

3)

```
acc.set_balance(100)
```

4)

```
BankAccount.set_balance(acc, 100)
```

Your Answer: 2) ✓

```
acc.balance = 100
```

Correct Answer: 2)

```
acc.balance = 100
```

Q8. Which statement correctly retrieves the area of a Rectangle instance `rect`?

1)

```
rect.area()
```

2)

```
rect.get_area()
```

3)

```
rect.area
```

4)

```
Rectangle.calculate_area(rect)
```

Your Answer: 3) ✓

```
rect.area
```

Correct Answer: 3)

```
rect.area
```

Q9. How would you add two Employee instances, `<code>emp1</code>` and `<code>emp2</code>`?

1)

```
emp1.add(emp2)
```

2)

```
emp1 + emp2
```

3)

```
emp1.__sum__(emp2)
```

4)

```
add(emp1, emp2)
```

Your Answer: 2) ✓

```
emp1 + emp2
```

Correct Answer: 2)

```
emp1 + emp2
```

Q10. How does the Dog class inherit the species attribute from the Animal class?

1)

```
Dog.species = self.species
```

2)

```
super().__init__("Dog")
```

3)

```
Animal.__init__(self, "Dog")
```

4)

```
self.species = Animal.species
```

Your Answer: 2) ✓

```
super().__init__("Dog")
```

Correct Answer: 2)

```
super().__init__("Dog")
```

Q11. How would you create an instance of `Car` with a started engine?

1)

```
car = Car()  
car.start_engine()
```

2)

```
engine = Engine()  
car = Car(engine)  
car.engine.start()
```

3)

```
car = Car()  
car.engine.start()
```

4)

```
engine = Engine()  
engine.start()
```

Your Answer: 2) ✓

```
engine = Engine()  
car = Car(engine)  
car.engine.start()
```

Correct Answer: 2)

```
engine = Engine()  
car = Car(engine)  
car.engine.start()
```

Q12. How would you check if an employee belongs to the `HR` department?

1)

```
if emp1.department == 'HR':
```

2)

```
if emp1.department.name == 'HR':
```

3)

```
if emp1.is_in_department('HR'):
```

4)

```
if emp1.department in ['HR', 'Human Resources']:
```

Your Answer: 2) ✓

```
if emp1.department.name == 'HR':
```

Correct Answer: 2)

```
if emp1.department.name == 'HR':
```

Q13. What will happen when you try to create an instance of `Circle` without defining the area method?

- 1) It will raise a `TypeError`.
- 2) It will create an instance without any issues.
- 3) It will create an instance but with a warning.
- 4) It will create an instance and automatically define the area method.

Your Answer: 1) ✓

It will raise a `TypeError`.

Correct Answer: 1)

It will raise a `TypeError`.

Q14. What will be the result of calling the `speak` method on an instance `gc**` of `Grandchild`?**

- 1) "Parent speaking"
- 2) "Child speaking"
- 3) It will raise an `AttributeError`.
- 4) It will raise an ambiguity error.

Your Answer: 2) ✓

"Child speaking"

Correct Answer: 2)

"Child speaking"

Q15. How would you check if two `Point2D` instances, `p1` and `p2`, are equal?

1)

```
p1.equals(p2)
```

2)

```
p1 == p2
```

3)

```
Point2D.equals(p1, p2)
```

4)

```
compare(p1, p2)
```

Your Answer: 2) ✓

```
p1 == p2
```

Correct Answer: 2)

```
p1 == p2
```

Q16. How would you use the `celsius_to_fahrenheit` method to convert a temperature, handling potential errors?

1)

```
temp_conv.celsius_to_fahrenheit(-300)
```

2)

```
try:
    temp_conv.celsius_to_fahrenheit(-300)
except ValueError as e:
    print(f"Error: {e}")
```

3)

```
convert_temperature(temp_conv, -300)
```

4)

```
TemperatureConverter.convert(-300)
```

Your Answer: 2) ✓

```
try:
    temp_conv.celsius_to_fahrenheit(-300)
except ValueError as e:
    print(f"Error: {e}")
```

Correct Answer: 2)

```
try:
    temp_conv.celsius_to_fahrenheit(-300)
except ValueError as e:
    print(f"Error: {e}")
```


Q17. How does polymorphism manifest in the given code with the `Shape`, `Circle`, and `Rectangle` classes?

- 1) It allows direct instantiation of the Shape class.
- 2) It requires explicit casting when using methods from the Shape class.
- 3) It ensures that all subclasses have the same attributes.
- 4) It enables the use of different methods with the same name, area, across different classes.

Your Answer: 4) ✓

It enables the use of different methods with the same name, area, across different classes.

Correct Answer: 4)

It enables the use of different methods with the same name, area, across different classes.

Q18. Consider the following script saved as `example_module.py`: What happens when you run this script from the command line?

- 1) It will raise an ImportError.
- 2) The script will execute the `multiply` function and print the result.
- 3) It will print an error stating `__main__` is not defined.
- 4) The script will not execute the `multiply` function unless imported as a module.

Your Answer: 2) ✓

The script will execute the `multiply` function and print the result.

Correct Answer: 2)

The script will execute the `multiply` function and print the result.

Q19. How do you import a module named `example_module` in Python?

- 1)

import example_module
- 2)

use example_module
- 3)

include example_module
- 4)

require example_module

Your Answer: 1) ✓

import example_module

Correct Answer: 1)

```
import example_module
```

Q20. Assuming a function `multiply` is defined in a module named `calculator`. How would you import and use this function in your Python script?

1)

```
import calculator

result = calculator.multiply(3, 4)
```

2)

```
from calculator import multiply

result = multiply(3, 4)
```

3)

```
import calculator as calc

result = calc.multiply(3, 4)
```

4)

All of the above

Your Answer: 2) ❌

```
from calculator import multiply

result = multiply(3, 4)
```

Correct Answer: 4)

All of the above

Q21. How do you open a file named `example.txt` in read mode in Python?

1)

```
open("example.txt", "w")
```

2)

```
open("example.txt", "a")
```

3)

```
open("example.txt", "r")
```

4)

```
open("example.txt", "x")
```

Your Answer: 3) ✓

```
open("example.txt", "r")
```

Correct Answer: 3)

```
open("example.txt", "r")
```

Q22. Assuming `data.txt` contains text data, how would you read the contents from the file?

1)

```
read_file('data.txt')
```

2)

```
open('data.txt').read()
```

3)

```
with open('data.txt', 'r') as file:  
    content = file.read()
```

4)

```
content = open_file('data.txt')
```

Your Answer: 3) ✓

```
with open('data.txt', 'r') as file:  
    content = file.read()
```

Correct Answer: 3)

```
with open('data.txt', 'r') as file:  
    content = file.read()
```

Q23. What method is used to read the entire contents of a file in Python?

1)

```
readline()
```

2)

```
readlines()
```

3)

```
read()
```

4)

```
readfile()
```

Your Answer: 3) ✓

```
read()
```

Correct Answer: 3)

```
read()
```

Q24. How would you load a JSON string into a Python dictionary?

1)

```
json.parse('{ "key": "value" }')
```

2)

```
json.loads('{ "key": "value" }')
```

3)

```
json.load('{ "key": "value" }')
```

4)

```
loads('{ "key": "value" }')
```

Your Answer: 2) ✓

```
json.loads('{ "key": "value" }')
```

Correct Answer: 2)

```
json.loads('{ "key": "value" }')
```

Q25. Consider the following code snippet: What does the `get_quote` function do?

1) Retrieves a random quote from a local text file.

2) Sends a request to a remote API to get a random quote and its author.

3) Fetches a predefined quote stored in the code.

4) Extracts quotes from a database using SQL queries.

Your Answer: 2) ✓

Sends a request to a remote API to get a random quote and its author.

Correct Answer: 2)

Sends a request to a remote API to get a random quote and its author.

Q26. How do you establish a connection to a PostgreSQL database using `psycopg2` in Python?

1)

```
conn = psycopg2.connect("dbname=mydatabase user=myuser password=mypassword")
```

2)

```
conn = psycopg2.connection("mydatabase", "myuser", "mypassword")
```

3)

```
conn = psycopg2.create_connection("dbname=mydatabase user=myuser password=mypassword")
```

4)

```
conn = psycopg2.open_connection("mydatabase", "myuser", "mypassword")
```

Your Answer: 1) ✓

```
conn = psycopg2.connect("dbname=mydatabase user=myuser password=mypassword")
```

Correct Answer: 1)

```
conn = psycopg2.connect("dbname=mydatabase user=myuser password=mypassword")
```

Q27. How do you retrieve the results of a query using `psycopg2`?

1)

```
result = cursor.fetch()
```

2)

```
result = cursor.fetchall()
```

3)

```
result = connection.fetch()
```

4)

```
result = connection.fetchall()
```

Your Answer: (Not Attempted)

Correct Answer: 2)

```
result = cursor.fetchall()
```

