1. What is OOP?

a. Object-Oriented Programming

b. Object-Oriented Protocol

c. Object-Oriented Parser

d. Object-Oriented Python

Answer: a

1. Which of the following is not an OOP concept in Python?

a. Inheritance

b. Encapsulation

c. Polymorphism

d. None of the above

Answer: d

1. Which keyword is used to create a class in Python?

a. def

b. class

c. create

d. new

Answer: b

1. Which keyword is used to create an instance of a class in Python?

a. def

b. create

c. instance

d. None of the above

Answer: d

1. Which of the following is not a fundamental principle of OOP?

a. Encapsulation

b. Polymorphism

c. Abstraction

d. Efficiency

Answer: d

1. What is the purpose of encapsulation in OOP?

a. To make objects more efficient

b. To restrict access to certain methods and attributes

c. To enable inheritance

d. None of the above

Answer: b

1. Which of the following is not an access modifier in Python?

a. Public

b. Protected

c. Private

d. Restricted

Answer: d

1. Which access modifier in Python is used to restrict access to methods and attributes to within the same class?

a. Public

b. Protected

c. Private

d. Restricted

Answer: c

1. Which access modifier in Python is used to restrict access to methods and attributes to within the same class and its subclasses?

a. Public

b. Protected

c. Private

d. Restricted

Answer: b

1. Which access modifier in Python is used to allow unrestricted access to methods and attributes?

a. Public

b. Protected

c. Private

d. Restricted

Answer: a

1. What is inheritance in OOP?

a. A way to reuse code and create new classes based on existing ones

b. A way to hide implementation details

c. A way to restrict access to certain methods and attributes

d. None of the above

Answer: a

1. Which keyword is used to inherit from a class in Python?

a. extend

b. inherit

c. base

d. None of the above

Answer: d

1. What is polymorphism in OOP?

a. A way to reuse code and create new classes based on existing ones

b. A way to hide implementation details

c. A way to have multiple methods with the same name but different implementations

d. None of the above

Answer: c

1. What is method overriding in Python?

a. A way to create new methods based on existing ones

b. A way to hide implementation details

c. A way to have multiple methods with the same name but different implementations

d. None of the above

Answer: c

1. What is method overloading in Python?

a. A way to create new methods based on existing ones

b. A way to hide implementation details

c. A way to have multiple methods with the same name but different implementations

d. None of the above

Answer: d

1. What is abstraction in OOP?

a. A way to reuse code and create new classes based on existing ones

b. A way to hide implementation details

c. A way to have multiple methods with the same name but different implementations

d. None of the above

Answer: b

1. What is the difference between a class and an object in Python?

a. A class is a blueprint for creating objects, while an object is an instance of a class.

b. A class is a collection of methods, while an object is a collection of variables.

c. A class is a collection of variables, while an object is a collection of methods.

d. None of the above.

Answer: a

1. What is the difference between encapsulation and abstraction in Python?

a. Encapsulation is the process of hiding implementation details, while abstraction is the process of creating a simplified representation of a complex system.

b. Encapsulation is the process of creating a simplified representation of a complex system, while abstraction is the process of hiding implementation details.

c. Encapsulation and abstraction are the same thing in Python.

d. None of the above.

Answer: a

1. What is the purpose of the str() method in Python?

a. To convert an object to a string representation.

b. To compare two objects for equality.

c. To initialize the instance variables of an object.

d. None of the above.

Answer: a

1. What is the purpose of the @property decorator in Python?

a. To define a getter method for a class variable.

b. To define a setter method for a class variable.

c. To define a method that can be called on an instance of a class.

d. None of the above.

Answer: a

1. What is a metaclass in Python?

a. A class that creates instances of other classes.

b. A class that is used to define the behavior of other classes.

c. A class that is used to override the default behavior of Python.

d. None of the above.

Answer: b