**Module 11. Classes and Objects**

**TRUE/FALSE**

1. In object-oriented programming, an object is an instance of a class.

Answer: T

1. The \_\_init\_\_ method is called automatically when a new object is created.

Answer: T

1. The self parameter in a class method refers to the class itself.

Answer: F

1. A class can contain both attributes and methods.

Answer: T

1. Encapsulation restricts direct access to some of an object’s components.

Answer: T

1. The \_\_str\_\_ method is used to define the string representation of an object.

Answer: T

1. In a UML diagram, the first section holds the list of the class's data attributes.

Answer: F

1. Object-oriented programming does not support code reuse.

Answer: F

1. Polymorphism allows objects to be treated as instances of their parent class.

Answer: T

1. Inheritance promotes code reuse by allowing new classes to inherit properties from existing classes.

Answer: T

1. The self parameter must be the first parameter of any method in the class.

Answer: T

1. The \_\_init\_\_ method initializes the object's attributes.

Answer: T

1. The bark method in the Dog class is an example of a constructor.

Answer: F

1. A class blueprint can create multiple objects with different attributes.

Answer: T

1. Abstraction in OOP allows defining what a class should do without specifying how it should do it.

Answer: T

1. The term “encapsulation” refers to hiding the object’s data attributes from outside the object.

Answer: T

1. In UML, the class diagram’s middle section lists the class’s methods.

Answer: F

1. The \_\_str\_\_ function is necessary for the object to be converted to a string.

Answer: F

1. Classes in OOP provide a way to structure and organize code into reusable pieces.

Answer: T

1. The method print is automatically called when you pass an object as an argument to the print function.

Answer: F

**MULTIPLE CHOICE**

1. What does the \_\_init\_\_ method do in a Python class?

a. Initializes the class itself

b. Creates new methods

c. Initializes the object's attributes

d. Defines the string representation of the object

Answer: C

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

a. class

b. def

c. object

d. init

Answer: A

1. What is encapsulation?

a. The practice of defining what a class should do without specifying how

b. The restriction of direct access to some of an object’s components

c. The ability of different classes to be treated as instances of their parent class

d. The reuse of code through inheritance

Answer: B

1. Which method is called automatically when an object is created?

a. \_\_str\_\_

b. \_\_init\_\_

c. new

d. del

Answer: B

1. Which of the following is a benefit of using classes and objects?

a. Less memory usage

b. Enhanced code security and integrity

c. Faster execution time

d. Reduced complexity in small programs

Answer: B

1. Which parameter refers to the current instance of a class?

a. self

b. this

c. cls

d. obj

Answer: A

1. In a UML class diagram, which section lists the class’s methods?

a. First section

b. Second section

c. Third section

d. Fourth section

Answer: C

1. Which method is used to define the string representation of an object in Python?

a. \_\_init\_\_

b. \_\_str\_\_

c. repr

d. del

Answer: B

1. What is polymorphism in object-oriented programming?

a. The ability to create new classes from existing classes

b. The practice of defining what a class should do without specifying how

c. The ability of different classes to be treated as instances of their parent class

d. The restriction of direct access to some of an object’s components

Answer: C

1. What does UML stand for?

a. Unified Modeling Language

b. United Modeling Language

c. Unified Model Language

d. Union of Modeling Languages

Answer: A

1. What is the purpose of the \_\_str\_\_ method in a class?

a. To initialize the object’s attributes

b. To create new methods

c. To define the string representation of the object

d. To restrict access to the object’s components

Answer: C

1. In a class diagram, where are the class’s data attributes listed?

a. First section

b. Second section

c. Third section

d. Fourth section

Answer: B

1. Which method is automatically called when an object is passed to the print function?

a. state

b. obj

c. \_\_str\_\_

d. \_\_init\_\_

Answer: C

1. Which of the following will create an object, worker\_joey, of the Worker class?

a. def\_\_init\_\_(worker\_joey):

b. class worker\_joey:

c. worker\_joey = Worker()

d. worker\_joey.Worker

Answer: C

1. What is a class in Python?

a. An instance of an object

b. A blueprint for creating objects

c. A method of an object

d. A data attribute of an object

Answer: B

1. Which principle of OOP promotes code reuse?

a. Encapsulation

b. Polymorphism

c. Inheritance

d. Abstraction

Answer: C

1. What does the self parameter allow you to do?

a. Access local variables

b. Access class-level variables

c. Access the attributes and methods of the object

d. Access the parent class

Answer: C

1. Which of the following is NOT a core principle of OOP?

a. Inheritance

b. Polymorphism

c. Encapsulation

d. Linear Execution

Answer: D

1. What does a UML class diagram help you visualize?

a. System workflows

b. Object interactions over time

c. System components, their relationships, and behaviors

d. User interactions with the system

Answer: C

1. What is an object in Python?

a. A blueprint for creating classes

b. An instance of a class

c. A function within a class

d. A data attribute within a class

Answer: B

1. What is the main difference between procedural programming and OOP?

a. OOP uses global variables, procedural does not

b. OOP focuses on functions, procedural focuses on objects

c. OOP organizes code into classes and objects, procedural uses a linear sequence of instructions

d. OOP is not suitable for large applications, procedural is

Answer: C

1. Which method is used to perform setup operations for an object?

a. setup

b. \_\_init\_\_

c. \_\_str\_\_

d. del

Answer: B

1. What is the purpose of the \_\_del\_\_ method in a class?

a. To initialize the object’s attributes

b. To create new methods

c. To define the string representation of the object

d. To clean up resources when an object is destroyed

Answer: D

1. Which of the following statements about classes and objects is true?

a. A class is an instance of an object

b. An object is an instance of a class

c. Classes cannot have methods

d. Objects cannot have attributes

Answer: B

1. Which section in the UML class diagram holds the name of the class?

a. First section

b. Second section

c. Third section

d. Fourth section

Answer: A