**Q1. What is the purpose of Python's OOP?**

* **Purpose**:
  + **Encapsulation**: Bundles data (attributes) and methods (functions) into a single unit (class).
  + **Abstraction**: Hides complex implementation details and exposes only necessary features.
  + **Inheritance**: Allows classes to inherit attributes and methods from other classes, promoting code reuse.
  + **Polymorphism**: Enables objects of different classes to be treated as objects of a common superclass.

**Q2. Where does an inheritance search look for an attribute?**

* **Inheritance Search**:
  + Python looks for an attribute in the following order:
    1. The **instance object** itself.
    2. The **class** of the instance.
    3. All **superclasses** of the class (in the order defined by the method resolution order, or MRO).

**Q3. How do you distinguish between a class object and an instance object?**

* **Class Object**:
  + A blueprint for creating instances.
  + Defined using the class keyword.
  + Example:

class MyClass:

pass

* **Instance Object**:
  + A specific object created from a class.
  + Created by calling the class.
  + Example:

instance = MyClass()

**Distinction**:

* A class object defines the structure and behavior.
* An instance object is a concrete realization of that structure.

**Q4. What makes the first argument in a class’s method function special?**

* **First Argument (self)**:
  + Refers to the instance of the class.
  + Allows methods to access and modify the instance's attributes.
  + Example:

class MyClass:

def my\_method(self):

print("Hello from", self)

**Q5. What is the purpose of the \_\_init\_\_ method?**

* **Purpose**:
  + A special method called when an instance of a class is created.
  + Used to initialize the instance's attributes.
  + Example:

class MyClass:

def \_\_init\_\_(self, value):

self.value = value

**Q6. What is the process for creating a class instance?**

* **Process**:
  1. Define the class.
  2. Call the class as if it were a function.
  3. Example:

class MyClass:

pass

instance = MyClass() # Create an instance

**Q7. What is the process for creating a class?**

* **Process**:
  1. Use the class keyword.
  2. Define attributes and methods.
  3. Example:

class MyClass:

def \_\_init\_\_(self, value):

self.value = value

def display(self):

print(self.value)

**Q8. How would you define the superclasses of a class?**

* **Superclasses**:
  + Classes from which a class inherits.
  + Defined in parentheses after the class name.
  + Example:

class SuperClass:

pass

class SubClass(SuperClass): # SuperClass is the superclass of SubClass

pass