**1. What is Object-Oriented Programming, and how does it differ from procedural programming?**

Object-Oriented Programming (OOP) is a programming paradigm that organizes code into objects, which are instances of classes. It focuses on encapsulation, inheritance, and polymorphism to structure code and promote reusability. In contrast, procedural programming relies on procedures or routines, treating the program as a series of steps executed sequentially.

**2. Explain the principles of OOP and how they are implemented in Python. Describe the concepts of encapsulation, inheritance, and polymorphism in Python**.

**Encapsulation**

Encapsulation involves bundling data and methods that operate on the data within a single unit, i.e., a class. In Python, you achieve encapsulation through class definitions and access modifiers like private and public.

**Inheritance**

Inheritance allows a class to inherit attributes and methods from another class. It promotes code reuse and establishes a hierarchy. In Python, you use the `class derived\_class(base\_class):` syntax to implement inheritance.

**Polymorphism**

Polymorphism allows objects of different classes to be treated as objects of a common base class. It can manifest as method overloading or method overriding. Python supports both, enabling flexibility in method definitions.

3. **What is the purpose of the self keyword in Python class methods?**

In Python, `self` is a convention referring to the instance of the class. It is the first parameter in every instance method and allows methods to access and modify instance attributes. It distinguishes instance variables from local variables.

4. **How does method overriding work in Python, and why is it useful?**

Method overriding occurs when a derived class provides a specific implementation for a method that is already defined in its base class. This allows for a more specialized behavior in the derived class. It is useful for refining or extending the functionality of the base class method.

5. **What is the difference between class and instance variables in Python?**

Class variables are shared among all instances of a class, while instance variables are unique to each instance. Class variables are defined outside any method in a class, whereas instance variables are defined within methods using `self`.

**6.** **Discuss the concept of abstract classes and how they are implemented in Python.**

Abstract classes cannot be instantiated and may contain abstract methods, which are declared but have no implementation. Python supports abstract classes through the `ABC` (Abstract Base Class) module and the `@abstractmethod` decorator.

**7. Explain the importance of the super() function in Python inheritance.**

`super()` is used to call a method from the parent class. It is essential for invoking the constructor or methods of the base class, enabling proper initialization and maintaining the inheritance hierarchy.

**8. How does Python support multiple inheritance, and what challenges can arise from it?**

Python supports multiple inheritance, allowing a class to inherit from more than one base class. Challenges may arise in case of method name conflicts or ambiguity, and the order in which base classes are inherited plays a crucial role.

**9. What is a decorator in Python, and how can it be used in the context of OOP?**

Decorators are functions that modify the behavior of other functions or methods. In OOP, decorators can be applied to methods to extend or modify their functionality. They enhance code readability and promote code reusability.

**10.** **What do you understand by Descriptive Statistics? Explain by Example.**

Descriptive statistics involve summarizing and presenting data to provide insights into its main features. Examples include measures such as mean, median, mode, and standard deviation, which describe the central tendency and dispersion of a dataset.

**11. What do you understand by Inferential Statistics? Explain by Example**

Inferential statistics involve making inferences or predictions about a population based on a sample of data. It includes techniques like hypothesis testing and confidence intervals to draw conclusions about the entire population from limited observations.