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

In Python, object-oriented Programming (OOPs) is a programming paradigm that uses objects and classes in programming. It aims to implement real-world entities like inheritance, polymorphisms, encapsulation, etc. in the programming.

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

An inheritance search looks for an attribute first in the instance object, then in the class, the instance was created from, then in all higher superclasses, progressing from left to right

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

The main difference between class objects and instance objects in Python is that class objects represent the class itself, while instance objects represent individual instances of the class.

Another difference is that class objects can have class-level attributes and methods that are shared among all instances of the class, while instance objects have their own set of attributes and methods that are independent of other instances of the same class.

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

Self, The keyword self represents the instance of a class and binds the attributes with the given arguments

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

A constructor of a [class in Python](https://www.mygreatlearning.com/academy/learn-for-free/courses/python-classes?gl_blog_id=77921) is defined using the \_\_init\_\_ method. The python \_\_init\_\_ is a reserved method in Python that behaves like any other member function of the class, except the statements written under its definition are used to initialize the data members of a class in Python, i.e. it basically contains assignment statements. This method is automatically called at the time of class instantiation or object creation.

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

To create instances of a class,  call the class using class name and pass in whatever arguments its \_\_init\_\_ method accepts.

Q7. What is the process for creating a class?

In Python, a class can be created by using the keyword class, followed by the class name.

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

A class that is derived from another class is called a subclass (also a derived class, extended class, or child class). The class from which the subclass is derived is called a superclass (also a base class or a parent class