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

**Answer:**

* **It makes code more reusable: OOP programs prevent you from repeating code because a class can be defined once and reused many times.**
* **Makes it easier to work with larger programs.**

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

**Answer:**

**All of these objects are namespaces (packages of variables), and the inheritance search is simply a search of the tree from bottom to top looking for the lowest occurrence of an attribute name. Code implies the shape of such trees.**

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

**Answer:**

**A class is a template for creating objects in program. It is a  logical entity A class does not allocate memory space when it is created.**

**Whereas,**

**The object is an instance of a class. It is a physical entity. Object allocates memory space whenever they are created.**

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

**Answer:**

**First Arguments in a class’s method acts as a pointer to the class.**

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

**Answer:**

**\_\_init\_\_ is basically a reserved method in python classes. It is called as a constructor in object-oriented terminology. This method is called when an object is created from a class and it allows the class to initialize the attributes of the class.**

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

**Answer:**

**Instance\_of\_class = class\_name()**

**This will create a class instance.**

Q7. What is the process for creating a class?

**Answer:**

**class className:**

**def \_\_init\_\_(Self):**

**pass**

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

**Answer:**

**class parent\_class:**

**def \_\_init\_\_(self,a):**

**self.a = a**

**pass**

**class child\_class(parent\_class):**

**def \_\_init(self,\*args):**

**super(child\_class,self).\_\_init\_\_(\*args)**