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

Python’s OOP provides means for declaring python classes and objects which lays the foundation for OOP concepts.

OOP concepts allows creation of structured program. It allows related properties and behaviors to be combined into classes from which individual objects can be created. Other feature called **inheritance** allows creation of one class inheriting properties of another class**.**

Using OOP in Python, the concept of **encapsulation** restricts access to methods and variables. This prevents data from direct modification.

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

An inheritance search looks for an attribute in the class tree from bottom to top. In this way, it looks for the lowest occurrence of the attribute name.

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

Class objects support two kinds of operations: attribute references and instantiation.

**class** **MyClass**:

*"""A simple example class"""*

i = 12345

**def** f(self):

**return** 'hello world'

The only operations understood by instance objects are attribute references. There are two kinds of valid attribute names: data attributes and methods.

A class attribute is an attribute that belongs to the class where as instance attribute belongs to individual objects(one instance of class). A class attribute is shared between all the objects and it is defined outside the constructor function. Each object has a separate instance attribute and it is defined inside the constructor.

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

The first variable in a class’s method is used to refer to the particular instance of that class, i.e., it acts as a handle to the instance being processed. Though it can be named as anything, common practice is to name it as self.

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

This method is called immediately after an instance of the class is created and it acts as the constructor method which contains the code for initializing all the instance attributes.

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

A class instance allows us to instantiate a class with required properties. The constructor of the class which is present inside \_\_init\_\_ method will get executed during this process.

<instance\_name> = <class\_name>([<constructor arguments>])

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

A class begins with ‘class’ keyword, followed by class name and a column. All the methods and code for that class is defined in an indented blocked below this line.

**class** **ClassName**:

<statement-1>

.

.

.

<statement-N>

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

While creating the child class, specify the name of the superclass inside paranthesis.

class child\_class\_name(<super\_class\_name>):

…….

**References:**

1. https://docs.python.org/3/tutorial/classes.html