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

Ans:

There are four major benefits to Python object-oriented programming:

**Encapsulation:** in python OOP, we bundle code into a single unit where we can determine the scope of each piece of data.

**Abstraction:** by using classes, we are able to generalize our object types, simplifying your program.

**Inheritance:** because a class can inherit attributes and behaviors from another class, we are able to reuse more code.

**Polymorphism:** one class can be used to create many objects, all from the same flexible piece of code.

Overall, python OOP is about code reuse, minimize redundancy and program by customizing what already exists instead of starting from scratch.

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

### Ans: 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 (by default). The search stops at the first place the attribute is found.

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

### Ans: **Classes are a kind of template for creating multiple instances. Classes also support operator overloading methods, which instances inherit, and treat any functions nested in the class as methods for processing instances.**

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

### Ans: **It always receives the instance object that is the implied subject of the method call. It’s usually called 'self' by convention.**

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

Ans: If the \_\_init\_\_ method is coded or inherited in a class, Python calls it automatically each time an instance of that class is created.

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

### Ans:

### A = class\_name() B = class\_name2(arg1, arg2)

### We write class name with parenthesis only if there is no argument in the –init()\_\_ method.

### We write class name with parenthesis and arguments inside the parenthesis

Q7. What is the process for creating a class?

Ans:

Class Student:

roll=5

def getRoll(self):

return self.roll

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

The class from which a class inherits is called the parent or superclass. A class which inherits from a superclass is called a subclass

Class Animal:

Def \_\_init\_\_(self,num\_of\_legs):

self.num\_of\_legs=num\_of\_legs

Class Dog(Animal):

Pass

Here Animal is the super class of Dog class.