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

**Ans:** Python was designed with OOP approach and it offers the following advantages: Provides a clear program structure and a clean code. Facilitates easy maintenance and modification of existing code. Since the class is sharable, the code can be reused.

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

**Ans:** Every time you use an expression of the form object. attr where object is an instance or class object, Python searches the namespace tree at and above object , for the first attr it can find. Because lower definitions in the tree override higher ones, inheritance forms the basis of specialization.

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

**Ans:** In simple words, Instance refers to the copy of the object at a particular time whereas object refers to the memory address of the class

object: An object is an instance of a class. All data members and member functions of the class can be accessed with the help of objects. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created), memory is allocated.

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

**Ans:** The first argument of every class method, including init, is always a reference to the current instance of the class. By convention, this argument is always named self. In the init method, self refers to the newly created object; in other class methods, it refers to the instance whose method was called.

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

**Ans:** The \_\_init\_\_ method is similar to constructors in C++ and Java. Constructors are used to initialize the object’s state. The task of constructors is to initialize(assign values) to the data members of the class when an object of class is created

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

**Ans:** When you create an object, you are creating an instance of a class, therefore "instantiating" a class. The new operator requires a single, postfix argument: a call to a constructor. The name of the constructor provides the name of the class to instantiate.

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

**Ans:**

1. Define the properties of the class, which include its type, how it stores data, and (for relationship classes) the relationship type.
2. *(optional)* Specify permissions. If you do not specify permissions for a class, BMC Atrium CMDB assigns default permissions.
3. Define one or more CI and relationship class attributes.
4. *(optional)* Propagate attributes in a weak relationship. This step is necessary only if you have created a relationship class that has a weak relationship in which the attributes from one class should be propagated to another class.
5. *(optional)* Specify indexes. Indexing can reduce database query time, so index attributes that you expect users to use in queries frequently.
6. *(optional)* Configure instance auditing for the class. Auditing enables you to track the changes made to instances of a class.

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

**Ans:** The class from which a class inherits is called the parent or superclass. A class which inherits from a superclass is called a subclass, also called heir class or child class. Superclasses are sometimes called ancestors as well.

A superclass is the class from which many subclasses can be created. The subclasses inherit the characteristics of a superclass. The superclass is also known as the parent class or base class.