Abu KAMARA

05 Feb. 2023 Answer Key

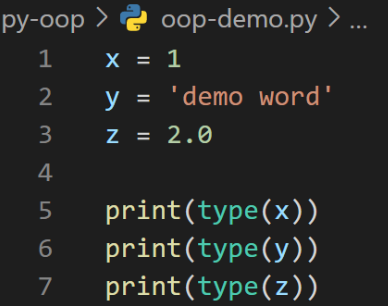
Q1. What are classes and objects?

In Python... everything is an object. This is because all objects are instances of a class.

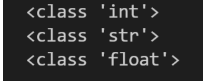
Okay, that is definitely confusing - which comes first, the object or the class?

The class. But why is that?

Classes are blueprints that describe the expected behaviour of an object. Every object is instantiated (created/built) from a class blueprint, even ones you might not think of!



Print()



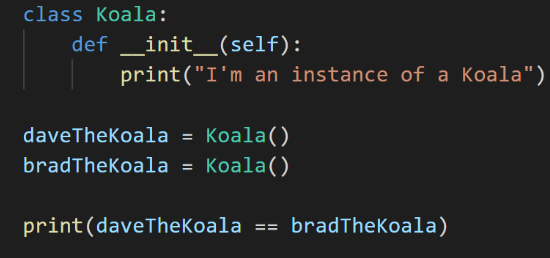
x, y, and z are all objects. And they are all instances of their respective data type class, meaning that under the hood Python has detected the expected behaviour of each of these variables and labelled (instantiated) them as their respective class. This means that we can do things like adding x to z to return 3.0, but we can't do things like x + y, because we will generate a type error:



But what is generating this error? It is all to do with the rules embedded within the blueprint that is behind the classes of integers and strings.

For each class, there is a ruleset within Python that prevents you from adding an integer and a string together, just as there is a rule that allows you to add an integer and a float together. Note that there is also a rule that if you were to add an integer and a float together that the end result will always be a float!

Instantiation is the act of creating an object using the blueprint of a class:



Debug:

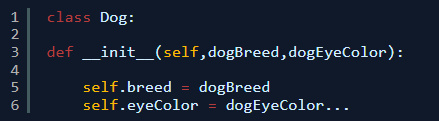


Q2. **The four pillars of OOP are:**

* Abstraction.
* Encapsulation.
* Inheritance.
* Polymorphism.

Q3.

The \_\_init\_\_ method is the Python equivalent of the C++ constructor in an object-oriented approach. The \_\_init\_\_ function is called every time an object is created from a class. The \_\_init\_\_ method lets the class initialize the object’s attributes and serves no other purpose. It is only used within classes.



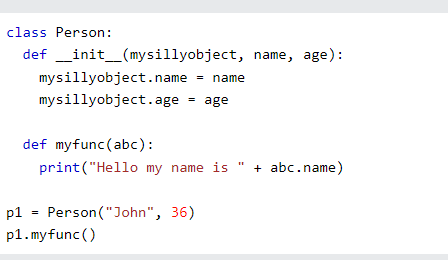
Q4. The self Parameter

The self-parameter is a reference to the current instance of the class and is used to access variables that belong to the class.

It does not have to be named self, you can call it whatever you like, but it has to be the first parameter of any function in the class:

Example

Use the words mysillyobject and abc instead of self:



**5. inheritance in Python**

One of the core concepts in object-oriented programming (OOP) languages is inheritance. It is a mechanism that allows you to create a hierarchy of classes that share a set of properties and methods by deriving a class from another class. Inheritance is the capability of one class to derive or inherit the properties from another class.

**The benefits of inheritance are:**

* It represents real-world relationships well.
* It provides the reusability of a code. We don’t have to write the same code again and again. Also, it allows us to add more features to a class without modifying it.
* It is transitive in nature, which means that if class B inherits from another class A, then all the subclasses of B would automatically inherit from class A.
* Inheritance offers a simple, understandable model structure.
* Less development and maintenance expenses result from an inheritance.