Question 1: Define Object Oriented Programming Language?

Object-Oriented programming is a widely used concept to write powerful applications. As a data scientist, you will be required to write applications to process your data, among a range of other things. In this tutorial, you will discover the basics of object-oriented programming in Python.

Question 2: List down the Benefits of OOP?

1. Modularity for easier troubleshooting

## 2. Reuse of code through inheritance

## 3. Flexibility through polymorphism

## 4. Effective problem solving

## Question 3: Differentiate between function and method?

1. Python method is called on an object, unlike a function. In our example above, we call start() on the object ‘car’. Conversely, we call Python function quite generically- we don’t call it on any object. Since we call a method on an object, it can access the data within it.
2. A method may alter an object’s state, but Python function usually only operates on it, and then prints something or returns a value.

## Question 4: Define the following terms:

## 1. Class

## 2. Object

## 3. Attribute

## 4. Behavior

Class definitions can appear anywhere in a program, but they are usually near the beginning (after the import statements). The syntax rules for a class definition are the same as for other compound statements. There is a header which begins with the keyword, class, followed by the name of the class, and ending with a colon.

This definition creates a new class called Point. The **pass** statement has no effect; it is only necessary because a compound statement must have something in its body. A docstring could serve the same purpose:

 syntax for selecting a variable from a module, such as math.pi or string.uppercase. Both modules and instances create their own namespaces, and the syntax for accessing names contained in each, called **attributes**, is the same. In this case the attribute we are selecting is a data item from an instance.