Question 1: Define Object Oriented Programming Language?

Object-Oriented Programming (OOP) is the term used to describe a programming approach based on objects and classes. The object-oriented paradigm allows us to organize software as a collection of objects that consist of both data and behavior. This is in contrast to conventional functional programming practice that only loosely connects data and behavior.

Question 2: List down the Benefits of OOP?

1. The programmes written with OOP are really easy to understand.
2. Since everything is treated as objects, so we can model a real-world concept using OOP.
3. OOP approach offers the reusability of classes. We can reuse the classes that are already created without writing them again and again.
4. Since the parallel development of classes is possible in OOP concept, It results in the quick development of the complete programmes.
5. Programmes written in OOP technique are marginally easier to test, manage as well as maintain.
6. It is a secured development technique since data is hidden and can’t be accessed by external functions.

Question 3: Differentiate between function and method?

A**function** is a piece of code that is called by name. It can be passed data to operate on (i.e. the parameters) and can optionally return data (the return value). All data that is passed to a function is explicitly passed.

A **method** is a piece of code that is called by a name that is associated with an object. In most respects it is identical to a function except for two key differences:

1. A method is implicitly passed the object on which it was called.
2. A method is able to operate on data that is contained within the class (remembering that an object is an instance of a class - the class is the definition, the object is an instance of that data).

Question 4: Define the following terms:

**Class:**

A class is a code template for creating objects. Objects have member variables and have behavior associated with them. In python a class is created by the keyword class.

**An Object:**

An object is an instance of a class. You could think of a class as the description of a concept, and an object as the realization of this description to create an independent distinguishable entity. For example, in the case of the Television, the class is the set of plans (or blueprints) for a generic television, whereas a television object is the realization of these plans into a real-world physical television. So there would be one set of plans (the class), but there could be thousands of real-world televisions (objects).

**Attribute:**

There are two types of attributes:

* An instance attribute is a Python variable belonging to one, and only one, object. This variable is only accessible in the scope of this object and it is defined inside the constructor function, \_\_init\_\_(self,..) of the class.
* A class attribute is a Python variable that belongs to a class rather than a particular object. It is shared between all the objects of this class and it is defined outside the constructor function, \_\_init\_\_(self,...), of the class.

**Behavior:**

Behavior-driven development (or BDD) is an agile software development technique that encourages collaboration between developers, QA and non-technical or business participants in a software project.