**Q1.Define Object Oriented Programming language?**

Answer: Object Oriented Programming (OOP) is a programming paradigm based on the concept of “Objects” which can contain “data”, in the form of fields (often known as attributes or properties), and code, in the form of procedures (often known as methods). A feature of objects is an object's procedures that can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another. OOP languages are diverse, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

**Q2. List down the Benefits of OOP?**

Answer: Object-oriented programming has several advantages over procedural programming:

* OOP is faster and easier to execute
* OOP provides a clear structure for the programs
* OOP helps to keep the C++ code DRY "Don't Repeat Yourself", and makes the code easier to maintain, modify and debug
* OOP makes it possible to create full reusable applications with less code and shorter development time

**Q3. Differentiate between function and method?**

Answer: 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).

**Q4.** **Define the following terms:**

1. **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.

1. **Object:**

An object is created using the constructor of the class. This object will then be called the instance of the class. In Python we create instances in the following manner

Instance = class(arguments)

1. **Attribute:**

Attributes are the individual things that differentiate one object from another and determine the appearance, state, or other qualities of that object. Let's create a theoretical class called Motorcycle. A motorcycle class might include the following attributes and have these typical values:

* Color: red, green, silver, brown
* Style: cruiser, sport bike, standard
* Make: Honda, BMW, Bultaco

Attributes of an object can also include information about its state; for example, you could have features for engine condition (off or on) or current gear selected.

Attributes are defined in classes by variables. Those variables' types and names are defined in the class, and each object can have its own values for those variables. Because each instance of a class can have different values for its variables, these variables are often called instance variables.

1. **Behavior:**

A class's behavior determines how an instance of that class operates; for example, how it will "react" if asked to do something by another class or object or if its internal state changes. Behavior is the only way objects can do anything to themselves or have anything done to them. For example, to go back to the theoretical Motorcycle class, here are some behaviors that the Motorcycle class might have:

* Start the engine
* Stop the engine
* Speed up
* Change gear
* Stall

To define an object's behavior, you create methods (Methods are functions defined inside classes that operate on instances of those classes.)