Q. Define Object Oriented Programming Language?

Object-oriented programming (OOP) is a programming language model in which programs are organized around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior. Examples of an object can range from physical entities, such as a human being that is described by properties like name and address, down to small computer programs, such as widgets.

Many of the most widely used programming languages (such as C++, Java, Python, etc.) are multi-paradigm and they support object-oriented programming to a greater or lesser degree

Q. List down the Benefits of OOP?

1. It provides a clear **modular structure** for programs which makes it good for defining abstract datatypes in which implementation details are hidden

2. Objects can also be **reused** within an across applications. The reuse of software also lowers the cost of development. More effort is put into the object-oriented analysis and design, which lowers the overall cost of development.

3. It makes software **easier to maintain**. Since the design is modular, part of the system can be updated in case of issues without a need to make large-scale changes

4. Reuse also enables **faster development**. Object-oriented programming languages come with rich libraries of objects, and code developed during projects is also reusable in future projects.

5. It provides a good framework for code libraries where the supplied software components can be **easily adapted and modified by the programmer**. This is particularly useful for developing graphical user interfaces.

6. **Better Productivity as OOP** techniques enforce rules on a programmer that, in the long run, help them get more work done.

Q. Differentiate between function and method?

'Method' is the object-oriented word for **'Function'.** However there is some difference in their calling convention and their usability.

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:

- A method is implicitly passed the object on which it was called.
- 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).

Q. Define the following terms:

1. Class:

A class is a blueprint for creating objects (a particular data structure), providing initial values for state (member variables or attributes), and implementations of behavior (member functions or methods).

2. Object:

Objects are the things you think about first in designing a program and they are also the units of code that are eventually derived from the process. In between, each object is made into a generic class of object and even more generic classes are defined so that objects can share models and reuse the class definitions in their code.

3. Attribute:

Attributes are data stored inside a class or instance and represent the state or quality of the class or instance. In short, attributes store information about the instance. Also, attributes should not be confused with class functions also known as methods.

4. 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 it’s internal state changes. Behavior is the only way objects can do anything to themselves or have anything done to them