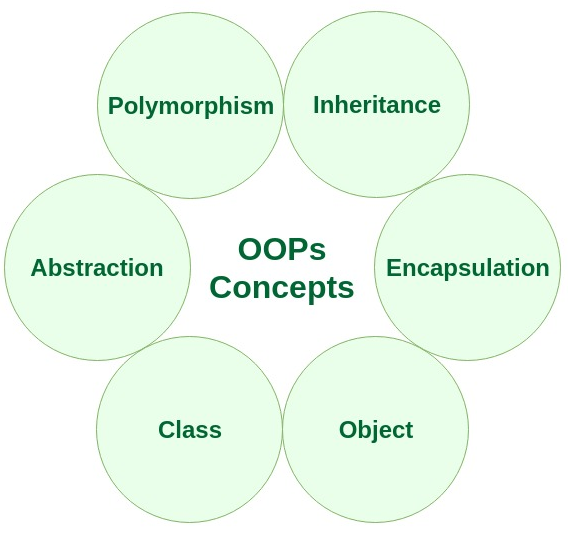
**Question 1: Define Object Oriented Programming Language?**

**Object-oriented programming:** As the name suggests, Object-Oriented Programming or OOPs refers to languages that uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

**OOPs Concepts:**

* Abstraction
* Class
* [Encapsulation](https://www.geeksforgeeks.org/encapsulation-in-java/)
* [Abstraction](https://www.geeksforgeeks.org/abstraction-in-java-2/)
* Information hiding
* Inheritance
* Interface
* Messaging
* Object
* Polymorphsim
* Procedure
* Method
* [Message Passing](https://www.geeksforgeeks.org/message-passing-in-java/)
* [**Abstraction**](https://www.webopedia.com/TERM/A/abstraction.html)**:** The process of picking out (abstracting) common features of objects and procedures.
* [**Class**](https://www.webopedia.com/TERM/C/class.html)**:** A category of objects. The class defines all the common properties of the different objects that belong to it.
* [**Encapsulation**](https://www.webopedia.com/TERM/E/encapsulation.html)**:** The process of combining elements to create a new entity. A procedure is a type of encapsulation because it combines a series of computer instructions.
* [**Information hiding**](https://www.webopedia.com/TERM/I/information_hiding.html)**:** The process of hiding details of an object or function. Information hiding is a powerful programming technique because it reduces complexity.
* [**Inheritance**](https://www.webopedia.com/TERM/I/inheritance.html)**:** a feature that represents the "is a" relationship between different classes.
* [**Interface**](https://www.webopedia.com/TERM/I/interface.html)**:** the languages and codes that the applications use to communicate with each other and with the hardware.
* [**Messaging**](https://www.webopedia.com/TERM/M/message_passing.html)**:** Message passing is a form of communication used in parallel programming and object-oriented programming.
* [**Object**](https://www.webopedia.com/TERM/O/object.html)**:** a self-contained entity that consists of both data and procedures to manipulate the data.
* [**Polymorphism**](https://www.webopedia.com/TERM/P/polymorphism.html)**:** A programming language's ability to process objects differently depending on their data type or class.
* [**Procedure**](https://www.webopedia.com/TERM/R/routine.html)**:** a section of a program that performs a specific task.
* [**Method**](https://www.geeksforgeeks.org/methods-in-java/)**:** A method is a collection of statements that perform some specific task and return result to the caller. A method can perform some specific task without returning anything. Methods allow us to **reuse** the code without retyping the code.
* [**Message Passing**](https://www.geeksforgeeks.org/message-passing-in-java/)**:** Objects communicate with one another by sending and receiving information to each other. A message for an object is a request for execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function and the information to be sent.



**Question 2: List down the Benefits of OOP?**

**1. Simplicity:** software objects model real world objects, so the complexity is reduced and the program structure is very clear;   
**2.** **Modularity:** each object forms a separate entity whose internal workings are decoupled from other parts of the system;   
**3.** **Modifiability:** it is easy to make minor changes in the data representation or the procedures in an OO program. Changes inside a class do not affect any other part of a program, since the only public interface that the external world has to a class is through the use of methods;   
**4.** **Extensibility:** adding new features or responding to changing operating environments can be solved by introducing a few new objects and modifying some existing ones;   
**5.** **Maintainability:** objects can be maintained separately, making locating and fixing problems easier;   
**6. Re-usability:** objects can be reused in different programs

**7.** **ENCAPSULATION:**wrapping of data and function into same entity so that it can be written once and can be used again anywhere in the whole program.

- The code can be easily modifiable.

- Easy to design user-friendly.

**Question 3: Differentiate between function and method?**

**Python Method**

1. Method is called by its name, but it is associated to an object (dependent).
2. A method is implicitly passed the object on which it is invoked.
3. It may or may not return any data.
4. A method can operate on the data (instance variables) that is contained by the corresponding class

**Functions**

1. Function is block of code that is also called by its name. (independent)
2. The function can have different parameters or may not have any at all. If any data (parameters) are passed, they are passed explicitly.
3. It may or may not return any data.
4. Function does not deal with Class and its instance concept.

**Difference between method and function**

1. Simply, function and method both look similar as they perform in almost similar way, but the key difference is the concept of ‘Class and its Object‘.
2. Functions can be called only by its name, as it is defined independently. But methods can’t be called by its name only, we need to invoke the class by a reference of that class in which it is defined, i.e. method is defined within a class and hence they are dependent on that class.

**Question 4: Define the following terms: 1. Class 2. Object 3. Attribute 4.Behavior.**

* [**Class**](https://www.webopedia.com/TERM/C/class.html)**:** A category of objects. The class defines all the common properties of the different objects that belong to it.
* [**Object**](https://www.webopedia.com/TERM/O/object.html)**:** a self-contained entity that consists of both data and procedures to manipulate the data.
* **Attribute:** An Attribute is a named property of a class. It has a type. It describes the range of values that property may hold.
* A value associated with an object which is referenced by name using dotted expressions. For example, if an object o has an attribute a it would be referenced as o.a.
* **Behavior:** It is represented by methods of an object. It also reflects the response of an object with other objects.

For Example: Cars have data like number of wheels, number of doors, seating capacity and also have behavior: accelerate, stop, show how much fuel are missing and so many other.

**Question 5: Write a code in python in which create a class named it Car which have 5 attributes such like (model, color and name etc.) and 3 methods. And create 5 object instance from that class.**

**Ans.** Jupyter note book file