1. What is OOP? List OOP concepts

Object-oriented programming (OOP) is **a style of programming characterized by the identification of classes of objects closely linked with the methods (functions) with which they are associated**. It also includes ideas of inheritance of attributes and methods.

As the name suggests, Object-Oriented Programming or OOPs refers to languages that use objects in programming, they use objects as a primary source to implement what is to happen in the code. Objects are seen by the viewer or user, performing tasks assigned by you. 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:

OOPS concepts are as follows:

1. Class
2. Object
3. Method
4. Abstraction
5. Encapsulation
6. Inheritance
7. Polymorphism

2. What is the difference between OOP and POP?

1. POP is procedure-oriented programming while OOP is object-oriented programming.
2. The main focus of POP is on "**how to get the task done**" it follows the flow chart to get the task done. OOP's main focus is on **data security** as only the objects of a class are allowed to access the attributes or function of a class.
3. The **functions** are small units of the large programs or a sub-program that execute to get the main task done. In contrast, OOP attributes and functions of the class are divided among the **objects**.
4. In POP, there is no specific accessing mode to access attributes or functions in the program. Conversely, in OOP there are three accessing modes "public", "private", "protected", that are used as an accessing method to access attributes or functions.
5. POP does not support the concept of Overloading/polymorphism. On the contrary, OOP supports Overloading/Polymorphism, which means using the same function name for performing different functions. We can overload functions, constructor, and operators in OOP.
6. There is no concept of inheritance in POP whereas, OOP supports inheritance which allows using the attribute and functions of other class by inheriting it.
7. POP is less secure as compared to OOP because in OOP the access specifier limits the access to attributes or functions which increase the security.
8. In POP if some data is to be shared among all the functions in the program, it is declared globally outside all functions. While in OOP the data member of the class can be accessed through the member functions of the class.
9. In POP there is no concept of the friend function. As against, in OOP there is a concept of friend function which is not the member of the class, but because it is friend member it can access the data member and member functions of the class.
10. There is no concept of virtual classes in POP whereas in OOP, the virtual functions support polymorphism.