Q. What is OOP? List OOP concepts .

Ans. Object-oriented programming (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

OOP focuses on the objects that developers want to manipulate rather than the logic required to manipulate them. This approach to programming is well-suited for programs that are large, complex and actively updated or maintained. This includes programs for manufacturing and design, as well as mobile applications; for example, OOP can be used for manufacturing system simulation software.

The organization of an object-oriented program also makes the method beneficial to collaborative development, where projects are divided into groups. Additional benefits of OOP include code reusability, scalability and efficiency.

The first step in OOP is to collect all of the objects a programmer wants to manipulate and identify how they relate to each other -- an exercise known as [data modeling](https://www.techtarget.com/searchdatamanagement/definition/data-modeling).

Examples of an object can range from physical entities, such as a human being who is described by properties like name and address, to small computer programs, such as [widgets](https://www.techtarget.com/whatis/definition/widget).

Once an object is known, it is labeled with a [class](https://www.techtarget.com/whatis/definition/class) of objects that defines the kind of data it contains and any logic sequences that can manipulate it. Each distinct logic sequence is known as a method. Objects can communicate with well-defined interfaces called messages.

List of OOPs Concepts

* Objects.
* Classes.
* Object.
* Class.
* Abstraction.
* Inheritance.
* Polymorphism.
* Encapsulation.

Q. What is the difference between OOP and POP?

Ans. The basic difference between OOP and POP both are programming methods, with OOP denoting “**Object-Oriented Programming**” and POP denoting “**Procedure Oriented Programming**.” OOP and POP both are high-level programming languages that solve problems, but they take distinct techniques. Programming paradigms are the technical terms for these methodologies. Because there is no one-size-fits-all solution to a problem, a programmer might use a variety of approaches while writing software.

This is where the use of oop and pop programming languages comes into play. A programme makes it simple to solve a problem by adopting the correct technique, or paradigm, as the case may be. Two such paradigms are object-oriented programming and procedure-oriented programming. A programming language is a set of rules and procedures that programmers use to give computers instructions to carry out. Every programming language has its own syntax, which allows you to tell a computer what tasks it should accomplish once you’ve learned it.

Computers are extremely powerful devices. With a computer, we can swiftly calculate numbers and create remarkable programmes for a variety of uses. To take advantage of this power, though, we must communicate with the computer in a less uncomfortable manner than manually entering ones and zeros.

As a result, we have programming languages that are supported by previously created machine code. The further we get away from machine code, the more abstract and specialized languages become in data management. This is why there are so many languages: no single language is ideal, and they all serve various and overlapping purposes.

Consider it this way: English is a language that allows you to communicate with other people who speak the same language as you. You can communicate with anyone who understands the basic norms of English if you know them. Computers, on the other hand, cannot understand English or any other “conventional” language.