

Key concepts of OOP:

The three pillars of object-oriented development are Encapsulation, Inheritance and Polymorphism. The key concepts of OOP are shown in below figure.

Objects:

Objects are the basic run-time entities in an object-oriented system. They may represent a person, a place, a bank account, a table of data or any item that the program must handle. Program objects should be chosen such that they match closely with the real-world objects. Objects take up space in the memory and have an associated address like structure in C. When a program is executed the objects interact by sending messages to one another.

Classes:

The entire set of data and code of an object can be made a user-defined data type with the help of a class. Objects are variable of type class. Once a class has been defined we can create any number of objects belonging to that class. A class is thus a collection of objects of similar type. Classes are user-defined data types and behave like the built-in types of programming language.

Encapsulation:

The wrapping up of data and functions into a single unit is known as encapsulation. The data is not accessible to the outside world and only those functions which are wrapped in the class can access it. The insulation of the data from direct access by the program is called data hiding or information hiding.

Data Abstraction:

Abstraction refers to the act of representing essential features without including the background details or explanations. Classes use the concept of abstraction and are defined as a list of abstract attributes such as size, weight and cost, and functions to operate on these attributes. Since the classes use the concept of data abstraction, they are known as Abstract Data Types.

Inheritance:

Inheritance is the process by which objects of one class acquire the properties of objects of another class. The principle behind this sort of division is that each derived class shares common characteristics with the class from which it is derived. In OOP the concept of inheritance provides the idea of reusability. This means that we can add additional features to an existing class without modifying it.

Polymorphism:

Polymorphism means the ability to take more than one form. That is an operation may exhibit different behaviours in different instances. The behaviour depends upon the types of data used in the operation. Example : operation of addition. For two numbers the operation will generate a sum. If the operands are strings, then operation would produce a third string by concatenation. The process of making an operator to exhibit different behaviours in different instances is known as Operator Overloading. Using a single function name to perform different types of tasks is known as function overloading.

Dynamic Binding:

Binding refers to the linking of a procedure call to the code to be executed in response to the call. Dynamic binding means that the code associated with a given procedure call is not known until the time of the call at run-time. It is associated with polymorphism and inheritance.

Message Communication:

An object oriented program consists of a set of objects that communicate with each other. Objects communicate with one another by sending and receiving information much that same way as people pass messages to one another. Message passing involves specifying the name of the object, the name of the function (message) and the information to be sent. emp.salary(emp\_name)