OOPS – Object Oriented Programming Structure

[https://www.webopedia.com/TERM/O/object_oriented_programming_OOP.html]

Object-oriented programming (OOP) refers to a type of computer programming (software design) in which programmers define not only the data type of a data structure, but also the types of operations (functions) that can be applied to the data structure.

In this way, the data structure becomes an object that includes both data and functions. In addition, programmers can create relationships between one object and another. For example, objects can inherit characteristics from other objects.

The following Webopedia definitions will help you better understand object-oriented programming:

Class

In object-oriented programming, a category of objects. For example, there might be a class called shape that contains objects which are circles, rectangles, and triangles. The class defines all the common properties of the different objects that belong to it.

Classes are templates used for defining new types. Classes describe both the properties and behaviors of objects. Properties contain the data that are exposed by the class. Behaviors are the functionality of the object and are defined by the public methods (also called member functions) and events of the class. Collectively, the public properties and methods of a class are known as the object interface. Classes themselves are not objects, but instead they are used to instantiate (i.e., create) objects in memory.

Object

Generally, any item that can be individually selected and manipulated. This can include shapes and pictures that appear on a display screen as well as less tangible software entities. In object-oriented programming, for example, an object is a self-contained entity that consists of both data and procedures to manipulate the data.

Abstraction

Abstraction is a process of hiding the implementation details and showing only functionality to the user. Abstraction lets you focus on what the object does instead of how it does. There are two ways to achieve abstraction in java.

1. Abstract class (0 to 100%)
2. Interface (100%)

Polymorphism

Generally, the ability to appear in many forms. In object-oriented programming, polymorphism refers to a programming language's ability to process objects differently depending on their data type or class. More specifically, it is the ability to redefine methods for derived classes. For example, given a base class shape, polymorphism enables the programmer to define different area methods for any number of derived classes, such as circles, rectangles and triangles. No matter what shape an object is, applying the area method to it will return the correct results. Polymorphism is considered to be a requirement of any true object-oriented programming language (OOPL).

Encapsulation

In programming, the process of combining elements to create a new entity. For example, a procedure is a type of encapsulation because it combines a series of computer instructions. Likewise, a complex data type, such as a record or class, relies on encapsulation. Object-oriented programming languages rely heavily on encapsulation to create high-level objects. Encapsulation is closely related to abstraction and information hiding. A class is the best example of encapsulation.

Inheritance

In object-oriented programing (OOP) inheritance is a feature that represents the "is a" relationship between different classes. Inheritance allows a class to have the same behavior as another class and extend or tailor that behavior to provide special action for specific needs.

Information hiding

In programming, the process of hiding details of an object or function. Information hiding is a powerful programming technique because it reduces complexity. One of the chief mechanisms for hiding information is encapsulation -- combining elements to create a larger entity. The programmer can then focus on the new object without worrying about the hidden details. In a sense, the entire hierarchy of programming languages, from machine languages to high-level languages, can be seen as a form of information hiding.

Information hiding is also used to prevent programmers from intentionally or unintentionally changing parts of a program.

---X---X---

Abstract class

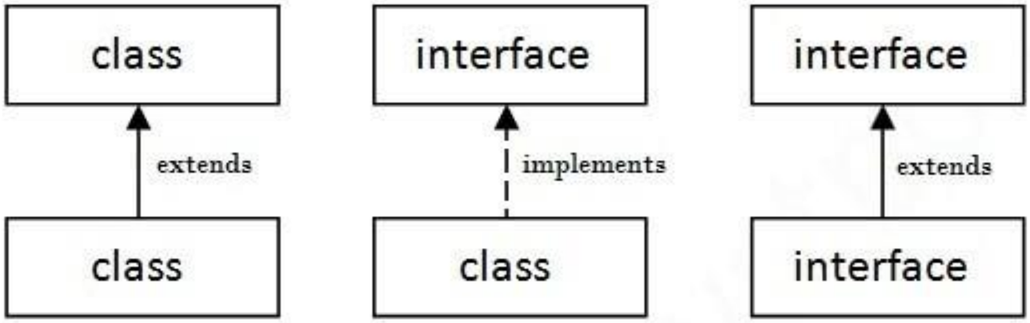
A class that is declared with abstract keyword is known as abstract class. It can contain one or more abstract method. An abstract method is a method that is only declared but not defined. For example, a method without any body as shown below:

Abstract void doSomething();   /* This method does not have a body i.e. no implementation.*/

Abstract class can only be extended and its method implemented. It cannot be instantiated.

Interface

An interface is a reference type in Java. It is similar to class. It is a collection of abstract methods. A class implements an interface, thereby inheriting the abstract methods of the interface. Along with abstract methods, an interface may also contain constants, default methods, static methods, and nested types.


class
interface
interface
extends
implements
extends
class
class
interface