

03/06/2024

Introduction to OOPS

- * OOP stands for Object-Oriented Programming
- * 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.
- * 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.
- * Procedural programming is about writing procedures or methods that perform operations on the data, while object-oriented programming is about creating objects that contain both data and methods.

Advantages of OOPs

- * OOP is faster and easier to execute
- * OOP provides a clear structure for the programs
- * OOP helps to keep the Java code "DRY" "Don't repeat yourself", and makes the code easier to maintain, modify and debug
- * OOP makes it possible to create full reusable applications with less code and shorter development time

Object → Basic Entities used to refer classes

class → set of objects (Data member, Data functions)

inheritance → parent to child class

polymorphism → Allows the object to behave in diff form

Abstraction → Hiding the implementation

Encapsulation → Wrapping of data members & function in single unit

Object:

* An object in Java is the physical as well as a logical entity (tangible and intangible). The example of an intangible object is the banking system.

* Example of a tangible object is a chair, pen, tablet it can be physical or logical.

* An object has three characteristics:

state: represents the data (value) of an object.

Behaviour: represents the behaviour (functionalities) of an object such as deposit, withdraw etc.

* Identity: An object identity is typically implemented via a unique ID.

class:

* A class is a group of objects which have common properties

* It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

* A class in Java can contain:

- (i) Fields
- (ii) Methods
- (iii) Constructors
- (iv) Blocks
- (v) Nested class and interface

Encapsulation:

* Encapsulation in Java is a process of wrapping code and data together into a single unit, for example, a capsule which is mixed of several medicines.

* The meaning of encapsulation, is to make sure that "sensitive" data is hidden from users.

*

* To achieve this, you must:

- * declare class variables/attributes as private
- * provide public get and set methods to access and update the value of a private variable.

Constructors in Java:

- * In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling constructor, memory for the object is allocated in the memory.
- * It is a special type of method which is used to initialize the object.
- * Every time an object is created using the new keyword, at least one constructor is called.
- * It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default.
- * Same name as class name
- * Special method used to initialize object.
- * Object created constructor is called.

This keyword in Java:

- * The this keyword refers to the current object in a method or constructor.
- * The most common use of this keyword is to eliminate the confusion between class attributes and parameters with the same name (because a class attribute is shadowed by a method or constructor parameter)
- * this can also be used to:
 - * Invoke current class constructor
 - * Invoke current class method
 - * Return the current class object
 - * Pass an argument in the method call
 - * Pass an argument in the constructor call