Objects and Classes in Java

An entity that has state and behavior is known as an object e.g., chair, bike, marker, pen, table, car, etc.

An object has three characteristics:

* **State:** represents the data (value) of an object.
* **Behavior:** represents the behavior (functionality) of an object such as deposit, withdraw, etc.
* **Identity:** An object identity is typically implemented via a unique ID. The value of the ID is not visible to the external user. However, it is used internally by the JVM to identify each object uniquely.

**An object is an instance of a class.** A class is a template or blueprint from which objects are created. So, an object is the instance(result) of a class.

**Object Definitions:**

* An object is *a real-world entity*.
* An object is *a runtime entity*.
* The object is *an entity which has state and behavior*.
* The object is *an instance of a 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.

**class** <class\_name>{

    field;

    method;

}

### Instance variable in Java

A variable which is created inside the class but outside the method is known as an instance variable. Instance variable doesn't get memory at compile time. It gets memory at runtime when an object or instance is created. That is why it is known as an instance variable.

### new keyword in Java

The new keyword is used to allocate memory at runtime. All objects get memory in Heap memory area.

## 3 Ways to initialize object

There are 3 ways to initialize object in Java.

1. By reference variable
2. By method
3. By constructor

## Anonymous object

Anonymous simply means nameless. An object which has no reference is known as an anonymous object. It can be used at the time of object creation only.

**new** Calculation();//anonymous object

1. Calling method through a reference:

Calculation c=**new** Calculation();

c.fact(5);

1. Calling method through an anonymous object

**new** Calculation().fact(5);