**Object Oriented Programming**

Java Object-Oriented Programming (OOP) is a way of programming that focuses on using objects. Instead of just writing a list of instructions, you create objects that represent real-world things, like a car or a person. This makes it easier to organize your code and model how things interact with each other, just like they do in real life.

**1.Class**

A class is like a blueprint. It defines what an object should be like, including its properties (like color or size) and behaviors (like running or jumping).

* **Example:** Imagine a Car class that describes what a car is, including properties like color and model.

**2. Class Properties**

Class properties are the characteristics of a class. They describe the state of an object created from the class.

* **Example:** In the Car class, properties could be color, model, and year.

**3. Objects**

An object is an instance of a class. It is a specific example of the class, with its own values for the properties defined by the class.

* **Example:** If Car is a class, then myCar is an object of that class, with its own color, model, and year.

**4. new Keyword**

The new keyword is used to create a new object from a class. It tells Java to make space in memory for the object.

* **Example:** Car myCar = new Car(); creates a new Car object.

**5. Object Creation**

When you create an object, you're making a specific instance of a class, with its own properties.

* **Example:** When you write new Car(); , you're creating a new car with default properties.

**6. instanceof Keyword**

The instanceof keyword checks whether an object belongs to a particular class.

* **Example:** myCar instanceof Car checks if myCar is actually a Car.

**7. Constructor**

A constructor is a special method that is called when you create a new object. It initializes the object's properties.

* **Example:** When you create a new Car with new Car("Red", "Toyota", 2022); , the constructor sets the color, model, and year of the car.