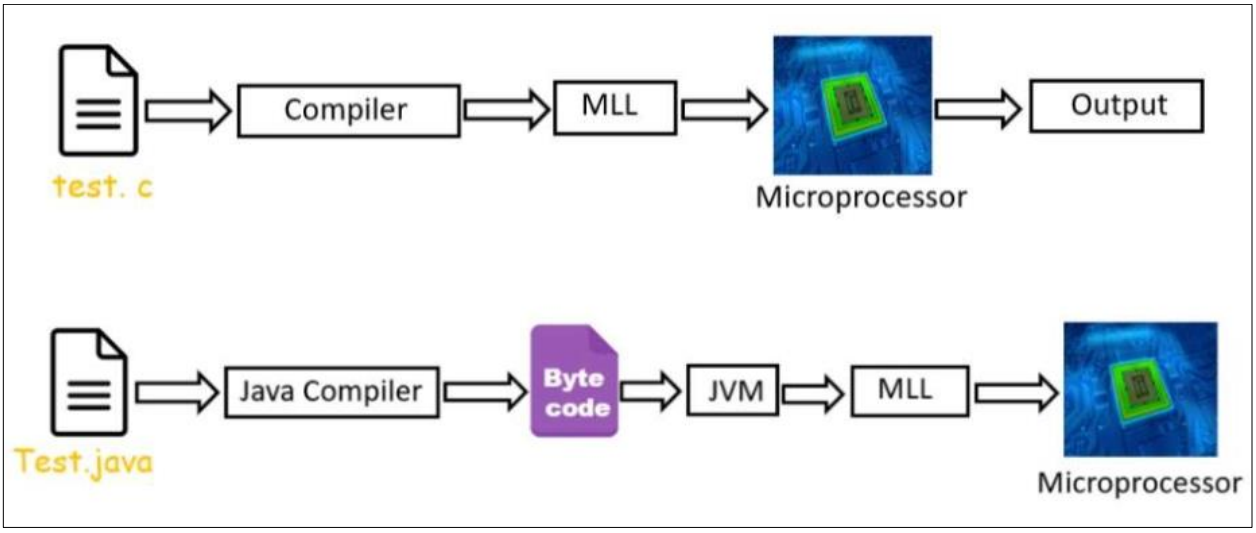
Object Orientation

Disadvantage of Java

- In java, extra step is involved in conversion from HLL to java code to MLL code java program are relatively slower in execution when compared to C/C++ programs.
- Bullet
- Time consumption is more in java.
- bullet
  Speed of execution is slower.



Flowchart 1 is execution steps in C and Flowchart 2 is execution steps in Java

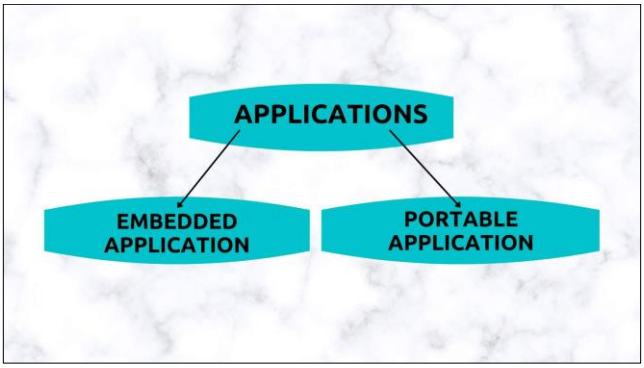
Introduction to Application

Applications

There are two types of applications

Embedded application - Speed of execution is at most important.

Portable application - Platform independence is at most important.



Embedded application

An embedded application is software that is placed permanently inside some kind of device to perform a very specific set of functions.

The program instructions for embedded systems are called firmware, or embedded software, and are stored in read-only memory, or flash memory chips.

For embedded applications Speed is very important factor**.**

Portable application

A portable application (portable app), sometimes also called standalone application, is a program designed to read and write its configuration settings into an accessible folder in the computer, usually in the folder where the portable application can be found.

This makes it easier to transfer the program with the user's preferences and data between different computers.

A program that doesn't have any configuration options can also be a portable application.

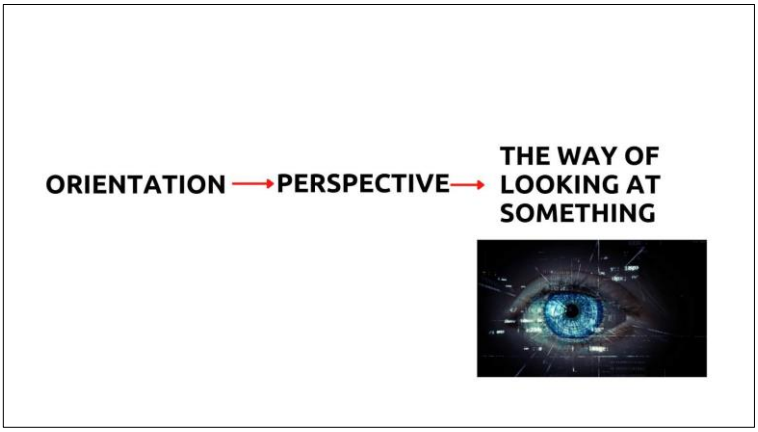
Portable applications can be stored on any data storage device, or in other words these applications are platform independent.

Object Orientation

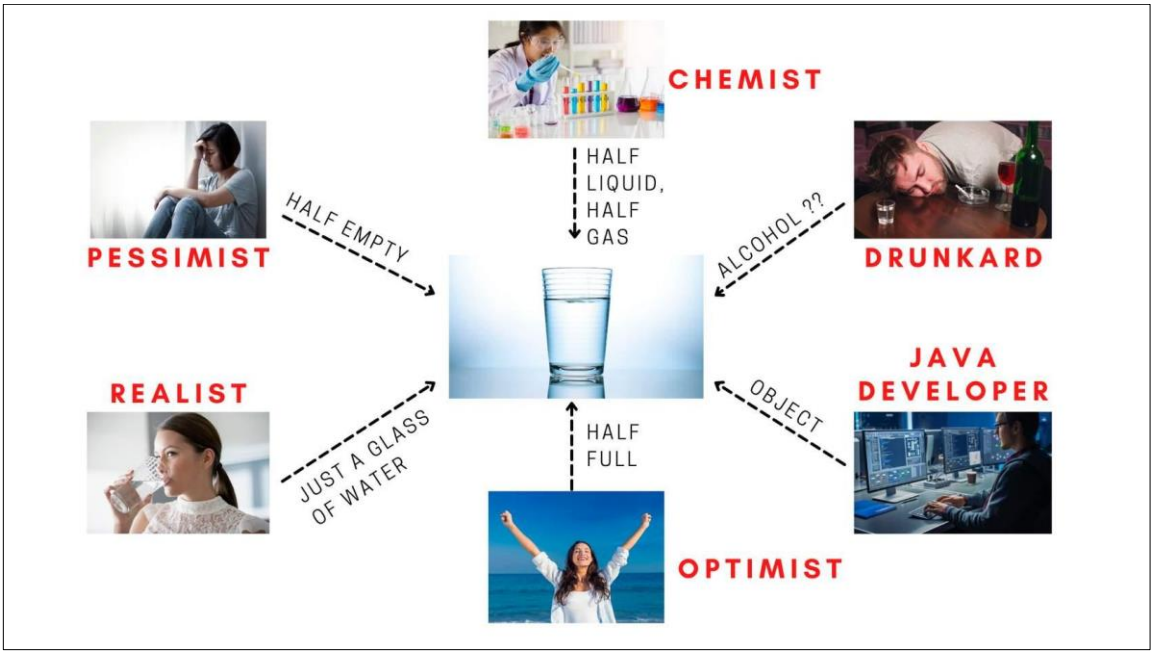
Object Orientation is the way of looking at this world as a collection of objects.

In this world no object is completely useless. All objects are in constant interaction with each other.

No object exists in isolation.



Let us take an example of glass of water and try to understand in better way:



Java become famous because of two main reasons: -

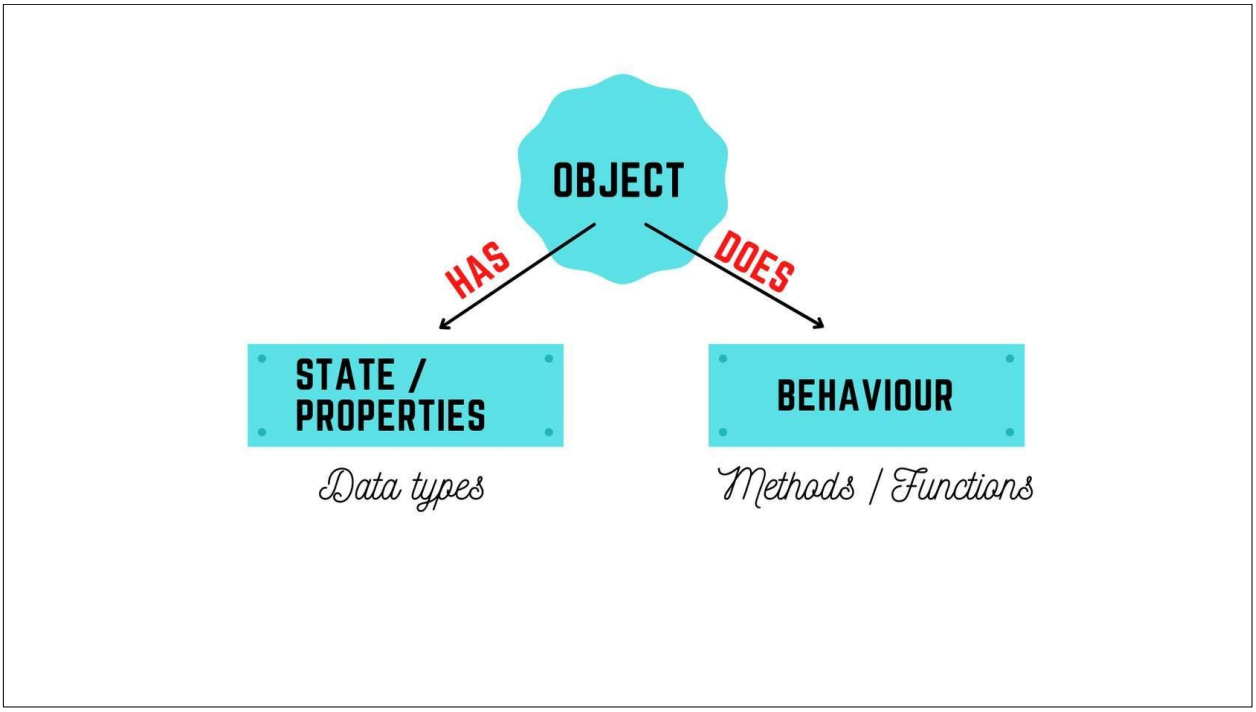
- Platform Independence.
- Object Orientation.

RULES OF OBJECT ORIENETATION

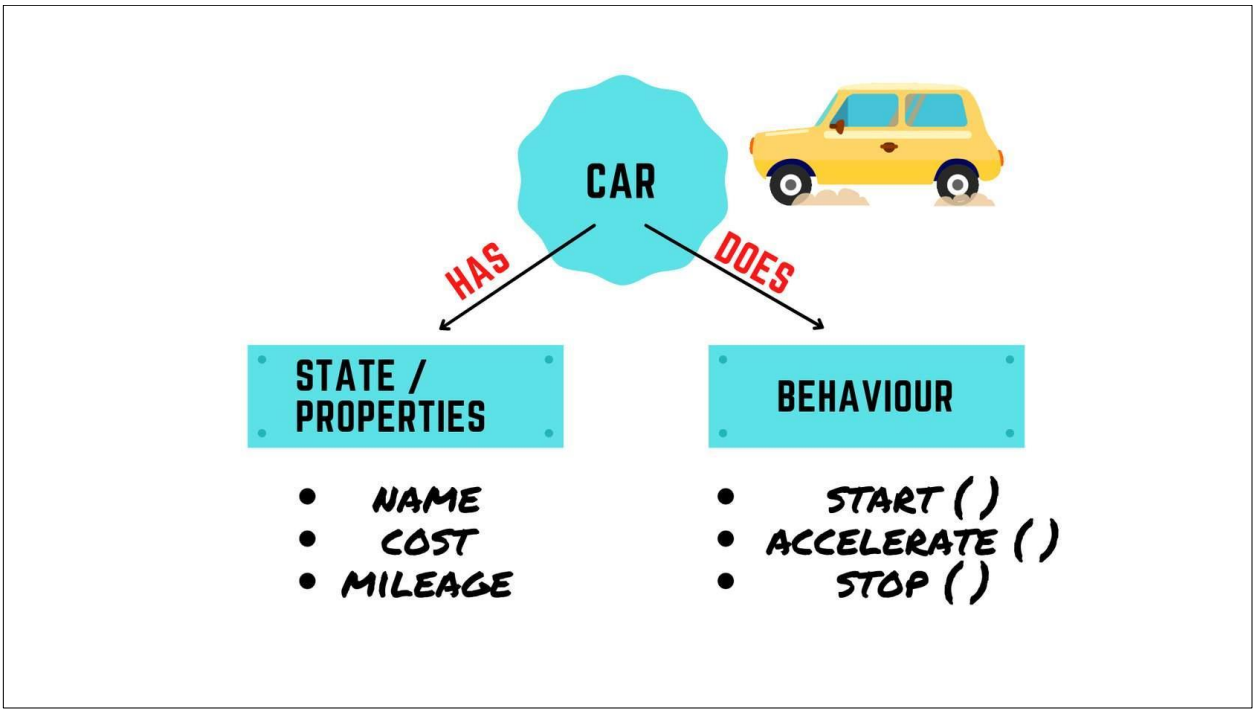
- The world is nothing but collection of objects.
- Every object in the world belong to type.

- In Java, every object has 2 parts i.e.,
  - State / properties means what an object has.
  - Behavior means what an object does.
- To take care of state we should use **datatypes**.
- To take care of behavior we should use **Methods / Functions.**

Examples for Object Orientation



EXAMPLE 1: **CAR OBJECT**



CODE

```
class Car
{
    //has part or state
     String name;
     int cost;
     float mileage;

     //does part or behavior
     void start()
     {
          //body of the method;
      }
      void accelerate()
      {
         //body of the method;
      }
       void stop()
       {
          //body of the method;
        }
}
```

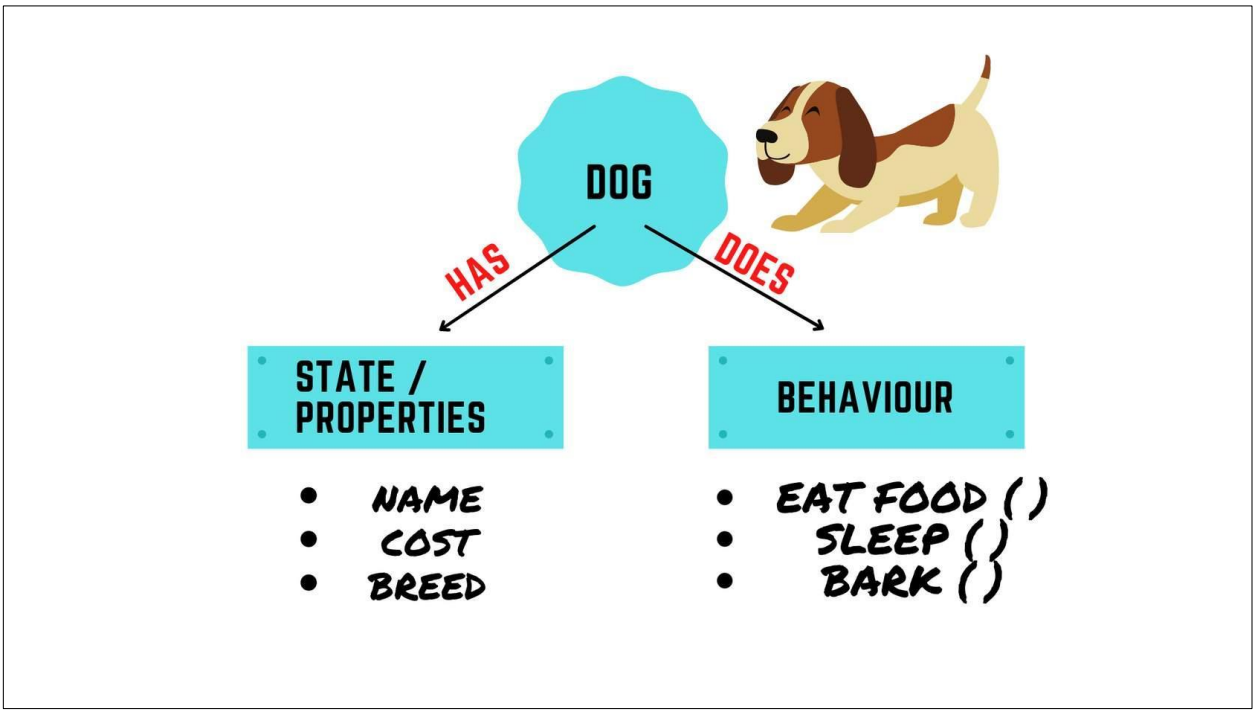
Let's see how to create an object in java:

```
Car c1 = new Car(); // object 1 creation
Car c2 = new Car(); // object 2 creation


c1.start(); //object 1 calling start ()
c1.accelerate(); //object 1 calling accelerate ()
c1.stop(); //object 1 calling stop ()


c2.start(); //object 2 calling start ()
c2.accelerate(); //object 2 calling accelerate ()
c2.stop(); //object 2 calling stop ()
```

EXAMPLE 2: **DOG OBJECT**



CODE

```
class Dog
{
    //has part or state
     String name;
     int cost;
     float breed;

     //does part or behavior
     void eatFood()
     {
          //body of the method;
      }
      void sleep()
      {
         //body of the method;
      }
       void bark()
       {
```

```
        //body of the method;
    }
}
```

Let's see how to create an object in java:

```
Dog d1 = new Dog(); // object 1 creation
Dog d2 = new Dog(); // object 2 creation


d1.eatFood(); //object 1 calling eatFood ()
d1.sleep(); //object 1 calling sleep ()
d1.bark(); //object 1 calling bark ()


d2.eatFood(); //object 2 calling eatFood ()
d2.sleep(); //object 2 calling sleep ()
d2.bark(); //object 2 calling bark ()
```