Object

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

- **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*.

For Example, Pen is an object. Its name is Reynolds; color is white, known as its state. It is used to write, so writing is its behavior.

What is a class in Java

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created.

Syntax to declare a class:

```
class <class_name>
{
   ………
   …….
```

```
    }
class Student
{
  Int id;
  String name;      // data member
 Void display()
{
sop
}  //member function
  public static void main (String args[])
    {
//Classname objectnm=new classname();
    Student s1=new Student();//creating an object of Student

    System.out.println(s1.id);//accessing member through reference variable

    System.out.println(s1.name);
    s1.display()
    }
}
```