

## Object Creation:

**Object:** Object is a real-world physical entity. Anything which is physically present, we consider it as an object.

- Object is nothing but instance of a class.

Object orientation is methodology to design a program by using classes and objects.

Every objects contains 2 characteristics:

1. State(represent data of an object)
2. Behaviour(represent behaviour of an object)

PEN(object):-

State:- name raynolds, color red etc.....

Behaviour:- used to write

**Object creation:** In java objects are created and managed by JVM. All the objects are created and stored in the heap memory.

Object address is always represented in hexa-decimal format.

There are 5 different ways to create the object in java.

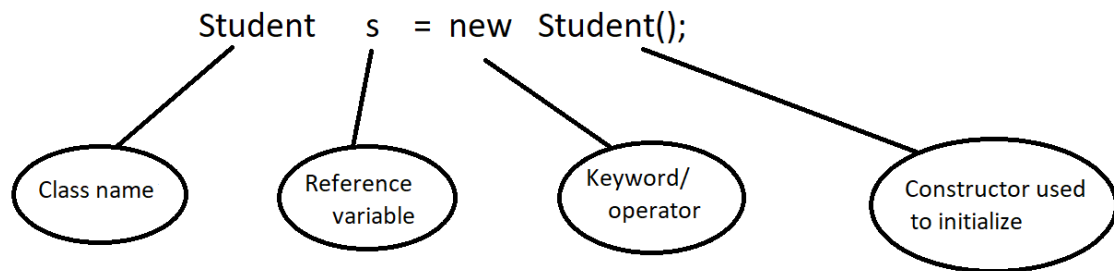
1. Using new Keyword
2. Using newInstance() method of the Class
3. Using newInstance() method of the Constructor
4. Using clone() method
5. Using Serialization and Deserialization

Using the **new** keyword is the most popular way to create an object or instance. When we create an object by using the new keyword, it allocates memory (heap) for the newly created **object** and also returns the **reference** of that object to that memory.

The syntax for creating an object is:

```
ClassName object = new ClassName();
```

Generalized example:

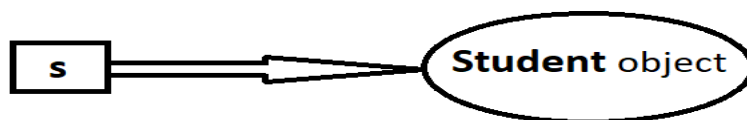


**Student s;**

- It notifies that **Student** is the class name with **s** the reference variable.
- Declaring reference variable doesn't mean creating object we must use new operator to create object.

**Student s = new Student();**

- The new operator instructs that a new object or a new memory location for the object must be created.
- The class name and the constructor name must be the same.
- The new operator creates object by allocating memory for new object and the reference variable pointing to that object.



Constructors are executed as part of the object creation. If we want perform any type of initializations at the time of object creation use constructors.

**Examples for object creation:**

```
class Student
{
String name;
int age;
float weight;
```

## JAVA

---

```
public static void main(String[] args)
{
    Student s = new Student();
    s.name="Dinga";
    s.age=20;
    s.weight=70.5f;
    System.out.println("The Student name is "+ s.name);
    System.out.println("The Student age is "+ s.age);
    System.out.println("The Student weight is "+ s.weight);
}
}
```

### Example2:

```
class Apple
{
    int kgs;
    float price;
    public static void main(String[] args)
    {
        Apple a = new Apple();
        a.kgs=1;
        a.price=120.0f;
        System.out.println("The apple is of kg"+ a.kgs);
        System.out.println("The apple price is "+ a.price);
    }
}
```



## JAVA

---

### Example3:

```
class Bike
{
String model = " R15v4 " ;
int price= 215000 ;
String color= " Black " ;
float mileage= 35.5f ;
public static void main(String[]args)
{
Bike b = new Bike();
System.out.println("The bike model is "+b.model);
System.out.println("The bike price is "+b.price);
System.out.println("The bike color is "+b.color);
System.out.println("The bike mileage is "+b.mileage);
}
}
```

### Example4:

```
Girlfriend g = new Girlfriend();
g.name="dingi";
g.height=5.5f;
g.weight=60.0;
g.mobNumber=9000000000l;
```

### Example5:

```
Pen p = new Pen();
p.name="Reynolds";
p.price=10;
```