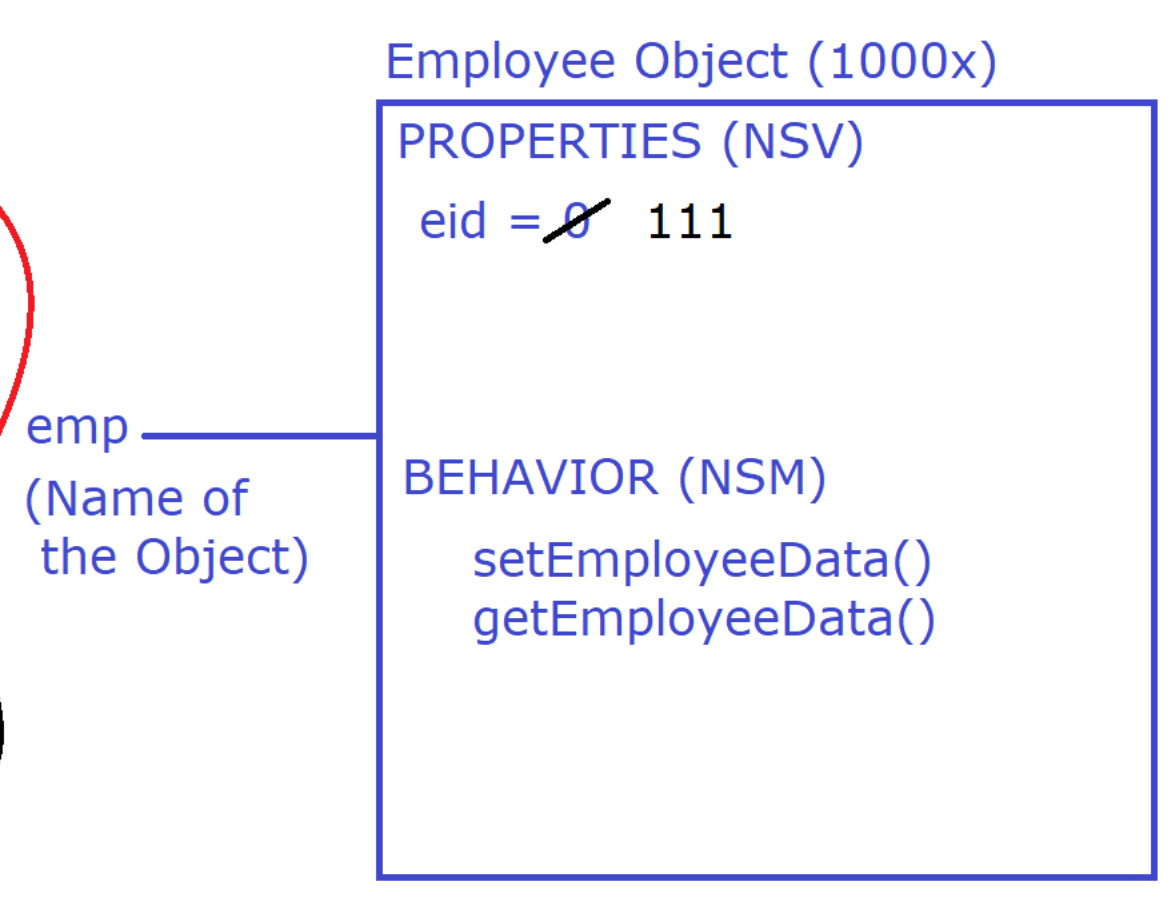
Constructor :

What is the advantage of writing Constructor in a class ?

```
public class Employee
{
    private int eid;

    public Employee() //Default no arg constructor
    {
    }

    public void setEmployeeData(int eid)
    {
       this.eid = eid;
    }
    public void getEmployeeData()
    {
       System.out.println(eid);
    }
}
 public class EmployeeDemo
 {
   public static void main(String [] args)
   {
      Employee emp = new Employee();
      emp.setEmployeeData(111);
   }
}
```



Note : In the above program non static variable initialization (eid = 0) is done at the time of creating the object with the new keyword and Java compiler, on the other hand the non static variable re-initiilization (eid = 111) is done at the time of calling setEmployeeData(int eid) method so, the conclusion is : Variable initialization and variable re-initialization both are done in two different lines.

If a user writes constructor :

```
public class Employee
{
    private int eid;

    public Employee(int eid) //User defined constructor
    {
       this.eid = eid;
    }

    public String toString()
    {
         return ""+this.eid;
    }
}
public class EmployeeDemo
{
   public static void main(String []args)
   {
     Employee emp = new Employee(111);
     System.out.println(emp);
   }
}
```

* If we write our own constructor then Variable initialization (eid = 0) and Variable re-initialization (eid = 111) BOTH ARE DONE AT THE SAME LINE AT THE TIME OF CREATING THE OBJECT.

* It is strongly recommended to initialize the non static variable with user value by using Constructor only.

So FINAL CONCLUSION IS :

1) To initialized the non static variable, We should always use Parameterized Constructor.
2) Methods are not a good choice to initialize the non static variable, Methods are mainly used to perform some operation on the Object.

Constructor Points :
----
* If the name of the class and name of the method both are exactly same and It does not contain any return type then it is called Constructor.
  Example

```
public class Test
{
   public Test()  //Constructor
   {
   }
}
```

* THE MAIN PURPOSE OF CONSTRUCTOR TO INITIALIZE THE NON STATIC VARIABLES i.e to initialize the object.

* A constructor never contain any return type including void also. If we provide return type then it will become method. We **can provide only return statement as shown in the program**

```
class Student
{
    public Student()
    {
        System.out.println("Constructor");
        return ; //Valid
    }
}
public class Test
{
    public static void main(String [] args)
    {
        new Student(); //Nameless OR Anonymous Object
    }
}
```

* Every java class must have al-least one constructor, either implicitly added by java compiler OR explicitly written by developer.

* A constructor is automatically called and executed at the time of creating the Object.

* Everytime we will create an object by using new keyword at-least one constructor must be invoked.

Types of Constructors :
----
* We have 3 types of constructors in java :

  1) Default No Argument Constructor [Added by java Compiler]
  2) No Argument OR Zero Argument OR Non Parameterized OR Parameter less constructor
  3) Parameterized Constructor

Default No Argument Constructor :
----
* If a constructor is added by java compiler in the class then it is **Default** no argument Constructor.
  Example :

```
public class Test          javac      public class Test
{                         ------>     {
                                          public Test()  //Default No Argument Constructor
}                                         {
                                          }
                                      }
```

2) No Argument OR Zero Argument OR Non Parameterized OR Parameter less constructor
----
If a constructor is written by user without argument then it is called No Argument OR Zero Argument OR non Parameterized OR parameter less constructor.

Example :

```
public class Student
{
    private int id;
    private String name;

    public Student() //Non parameterized constructor written by user
    {
       id = 0;
       name = null;
    }
}
```

Note : default no argument constructor (added by java compiler) and non parameterized constructor (written by user) both look like same the only difference is, If we don't write any constructor then by default one constructor is available called Default no argument constructor.

Parameterized Constructor :

If we pass one or more argument to the constructor then it is called parameterized constructor.

By using parameterized constructor all the objects will be initialized with different values.

Example :

```
public class Employee
{
   int id;
   String name;

   public Employee(int id, String name)
   {
     this.id = id;
     this.name = name;
   }
 }
```

```
package com.ravi.constructor;

//BLC
public class Dog
{
    private String name;
    private int age;
    private double height;
    private String color;


    public Dog(String name, int age, double height, String color)
    {
        super();
        this.name = name;
        this.age = age;
        this.height = height;
        this.color = color;
    }

    @Override
    public String toString()
    {
        return "Dog [name=" + name + ", age=" + age + ", height=" + height + ", color=" +
color + "]";
    }
}
```

```
package com.ravi.constructor;

//ELC
public class DogDemo
{
    public static void main(String[] args)
    {
        Dog d1 = new Dog("Tommy", 3, 3.5, "White");
        System.out.println(d1);

        System.out.println("............");
        Dog d2 = new Dog("Tiger", 5, 4.5, "Black");
        System.out.println(d2);

    }

}
```