



JAVA

Class 22

Agenda

Constructor

Types of Constructors

Inheritance in JAVA

Constructor

Object creation syntax:

Class-name reference-variable = new
classname();

Test t = new Test();

Test ---> class Name

t ---> Reference variables

= ---> assignment operator

new ---> keyword used to create object

Test() ---> constructor ;

; ---> statement terminator

Constructor

When we create instance (Object) of a class using new keyword, a constructor for that class is called.

- New keyword is used to create object in java.
- When we create object by using new operator after new keyword that part is constructor and it is executing a constructor.

Constructor

In Java, a constructor is a block of codes similar to the method.

It is called when an instance of the object is created, and memory is allocated for the object.

It is a special type of method which is used to initialize the object.

Rule: If there is no constructor in a class, compiler automatically creates a default constructor.

Rules to declare constructor

Rules to declare constructor:

1. Constructor name class name must be same.
1. It is possible to provide parameters to constructors (just like methods).
1. Constructor not allowed explicit return type even void.

Type of constructors

There are two types of constructors:

1. Default Constructor (provided by compiler)
1. User defined Constructor (provided by user) or parameterized constructor

```
Test.java
1 package com.Class12;
2
3 public class Test {
4
5     Test() {
6         System.out.println("I am default constructor");
7     }
8
9     Test(int a) {
10        System.out.println("I am parameterized constructor");
11    }
12 }
13
```

Default Constructor

- Inside the class if we are not declaring any constructor then compiler generates zero argument constructors with empty implementation at the time of compilation is called default constructor.
- The compiler generated constructor is called default constructor.
- Inside the class default constructor is invisible mode.
- The purpose of default constructor is to provide the default values to the object like 0, null etc. depending on the type.

Parameterized Constructor

- ▶ If any constructor contain list of variable in its signature is known as parameterized constructor. A parameterized constructor is one which takes some parameters.
- ▶ Parameterized constructor is used to provide different values to the distinct objects.

Syntax:

```
class ClassName {
```

```
    ClassName(list of parameters) { //parameterized constructor  
    }
```

```
}
```

Syntax to call parameterized constructor

```
ClassName objref=new ClassName(value1,  
value2,.....);
```

OR

```
new ClassName(value1, value2,.....);
```

Parameterized Constructor - Important points

- Whenever we create an object using parameterized constructor, it must be define parameterized constructor otherwise we will get compile time error.
- Whenever we define the objects with respect to both parameterized constructor and default constructor, It must be define both the constructors.
- In any class maximum one default constructor but 'n' number of parameterized constructors.

Example of Parameterized Constructor

```
class Test {  
    int a, b;  
    Test(int n1, int n2) {  
        SOP("I am from Parameterized Constructor...");  
        a=n1;  
        b=n2;  
        SOP("Value of a = "+a);  
        SOP("Value of b = "+b);  
    }  
};  
class TestDemo1 {  
    public static void main(String k []) {  
        Test t1=new Test(10, 20);  
    }  
};
```

Sample Program

```
3
4 public class Test {
5
6
7     void m1()
8     {
9         System.out.println("m1 method");
10    }
11    Test()
12    { System.out.println("0-arg constructor");
13    }
14    Test(int i)
15    { System.out.println("1-arg constructor");
16    }
17    public static void main(String[] args)
18    { Test t1=new Test();
19      Test t2=new Test(10);
20      t1.m1();
21      t2.m1();
22    }
23    }
24    //✓ The compiler generated 0-argument is called default constructor.
25    //✓ The user defined 0-argument constructor is not a default constructor.
26    //✓ The constructor arguments are local variables.
```

Problems @ Javadoc Declaration Console TestNG Debug

Terminated: Test [Java Application] C:\Program Files\Java\jdk-1.8.0_101\bin\javaw.exe (Aug 21, 2018, 11:27:47)

```
0-arg constructor
1-arg constructor
m1 method
m1 method
```

Sample Program

```
1 package Programs;
2
3
4 public class Test {
5
6
7     Test(int i)
8     {
9         System.out.println("1-arg constructor");
10    }
11
12    public static void main(String[] args)
13    {
14        Test t1=new Test(); //error : inside the class no 0-arg constructor
15
16        Test t2=new Test(10);
17    }}
18
19 /*If we are trying to compile below application the compiler will generate error message
20 "Cannot find symbol " because compiler is unable to generate default constructor.*/
```

Rules or properties of a constructor

1. Constructor will be called automatically when the object is created.
2. Constructor should not return any value even void also. Because basic aim is to place the value in the object. (if we write the return type for the constructor then that constructor will be treated as ordinary method).
3. Constructor definitions should not be static. Because constructors will be called each and every time, whenever an object is creating.
4. Constructor should not be private provided an object of one class is created in another class (Constructor can be private provided an object of one class created in the same class).
5. Constructors will not be inherited from one class to another class (Because every class constructor is create for initializing its own data members).

Task

1. Write a program that will have a constructor: one with parameters and second without any parameters. Create a separate Test class where you will execute both of the constructors.
2. Write a java program of Class Students that takes students name and 3 subject grades. Inside your class also have a method to Calculate Average Grade. Test Student class for 5 different students with different marks. Your program should print an average mark of each students name.

NOTE: please use different names for instance and local variables.

Task

1. Write a program that will have a private default constructor class and create 2 objects of this class: 1 - inside same class; 2 - from outside the class and observe result.
2. Write program that have static constructor and observe result.

Method vs Constructor

Method	Constructor
Method can be any user defined name	Constructor must be class name
Method should have return type	It should not have any return type (even void)
Method should be called explicitly either with object reference or class reference	It will be called automatically whenever object is created
Method is not provided by compiler in any case.	The java compiler provides a default constructor if we do not have any constructor.

Parameterized /User defined constructor

Constructors are used to write the logics these logics are executed during object creation.

Method also used to write the logics these logics are executed when we call the method.

Inheritance

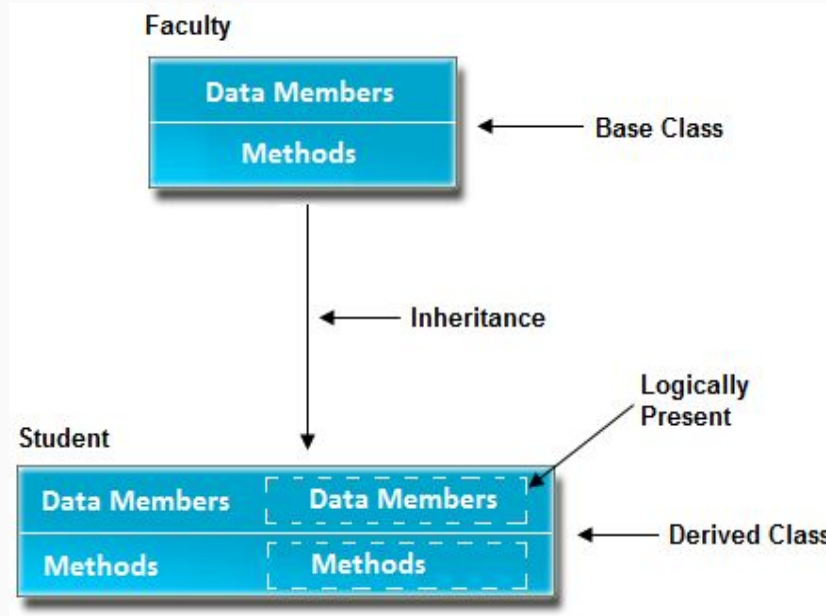
Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object.

The idea behind inheritance in java is that you can create new classes that are built upon existing classes.

When we inherit from an existing class, we can reuse methods and fields of parent class, and we can add new methods and fields also.

`extends` is the keyword used to inherit the properties of a class.

Inheritance



In the diagram data members and methods are represented in broken line are inherited from faculty class and they are visible in student class logically.

Inheritance

Inheritance represents the **IS-A** relationship, also known as parent-child relationship.

Syntax of Inheritance

```
class Subclass-Name extends Superclass-Name {  
    //methods and fields  
}
```

Advantage of Inheritance

If we develop any application using concept of Inheritance then that application have following advantages:

Application development time is less.

Application take less memory.

Application execution time is less.

Advantage of Inheritance

Code reusability: the same methods and variables which are defined in a parent/super/base class can be used in the child/sub/derived class.

Application performance is enhanced (improved).

Redundancy (repetition) of the code is reduced or minimized so that we get consistent results and less storage cost.

Note: In Inheritance the scope of access modifier increasing is allow but decreasing is not allow. Suppose in parent class method access modifier is default then it's present in child class with default or public or protected access modifier but not private(it decreased scope).

Disadvantage of Inheritance

The main disadvantage of using inheritance is that the two classes (parent and child class) get tightly coupled.

This means that if we change the code of parent class, it will affect to all the child classes which are inheriting/deriving the parent class, and hence, it cannot be independent of each other.