Chapter 9 – Constructors

Constructors :

A constructor is a special method which has same name of the class name and which has no return type.

We write constructors in Class

Constructor is a special kind of non-static method ( instance )

Mainly Constructors are used to initialize the non-static variable values

Types of Constructors

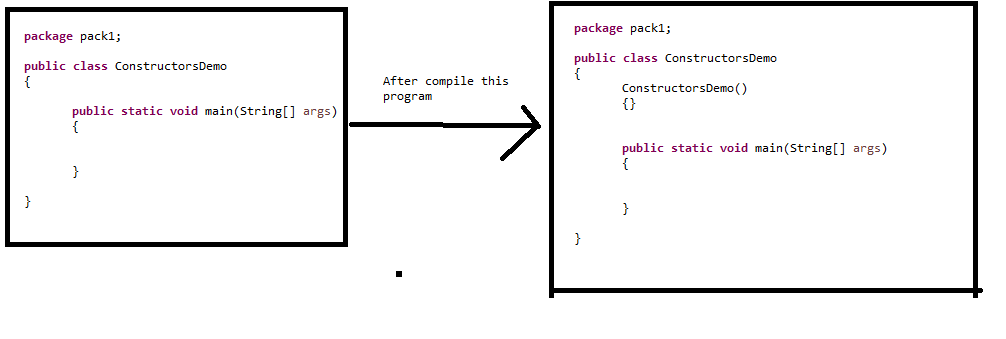
We have two types of Constructors

1. Default Constructors
2. User Defined Constructors

1.Default Constructors:

If the programmer does not defne any Constructor explicitly in the class, at the time of compile JVM will create a non-parameterized constructor in the class

Ex:



2.User defined Constructors

These are defined by programmers

Two types:

1. Without Parameters
2. With Parameters
3. **package** pack1;
4. **public** **class** ConstructorsDemo
5. {
6. ConstructorsDemo()
7. {
8. System.***out***.println("No parameter constructor");
10. }
12. ConstructorsDemo(**int** a,**int** b)
13. {
15. System.***out***.println("Sum of give values is "+(a+b));
16. }
18. ConstructorsDemo(**int** a)
19. {
20. System.***out***.println("Given number is "+a);
21. }
23. **void** display()
24. {
25. System.***out***.println("display method");
26. }
28. **public** **static** **void** main(String[] args)
29. {
30. ConstructorsDemo obj1=**new** ConstructorsDemo();
31. obj1.display();
32. obj1.display();
33. obj1.display();
34. ConstructorsDemo obj2=**new** ConstructorsDemo();
35. ConstructorsDemo obj3=**new** ConstructorsDemo(10,20);
36. ConstructorsDemo obj4=**new** ConstructorsDemo(100);
37. obj4.display();



42. }
43. }

To call constructors we have to create object.

If we want to execute constructors for 5 times then we have to create 5 objects.

To call a method we use object ref for instance methods.

To execute same instance method for 5 times, then we use the same ref variable.

Ex: Assigning instance variable values without constructors

**package** pack1;

**public** **class** Employee

{

String empName;

**int** empId;

**public** **static** **void** main(String[] args)

{

Employee emp1=**new** Employee();

emp1.empName="Lokesh";

emp1.empId=100;

System.***out***.println(emp1.empName);

System.***out***.println(emp1.empId);

Employee emp2=**new** Employee();

emp2.empName="Sreekar";

emp2.empId=101;

System.***out***.println(emp2.empName);

System.***out***.println(emp2.empId);

}

}

Assigning instance variables values with Constructors

**package** pack1;

**public** **class** Employee

{

Employee(String empName,**int** empId)

{

System.***out***.println(empName);

System.***out***.println(empId);

}

**public** **static** **void** main(String[] args)

{

Employee emp1=**new** Employee("Lokesh",100);

Employee emp2=**new** Employee("sreekar",101);

}

}

Constructors Overloading:

Same constructor name with different parameters