**Constructor :**

Constructor is used to initialize the object, but not for creating object.

**Rules to create Constructor :**

1. Class name and constructor name should be same.
2. Constructor doesn’t have any return type not even void.
   1. If we write program like below with return type as void for constructor, will execute or not?

Public class Demo{

Void Demo(){

}

Public static void main(String[] args){

Demo d1 = new Demo();

}

}

Ans: Will compile the code, because compiler treats that as normal method, but not considered as constructor. This kind of programming (method name exactly like class name) is not recommended.

3. Which modifiers applicable for Constructors?

Ans : public, private, protected and default

If we try to use any other modifier other than above mentioned modifiers we will get compile time error like (modifier <modifier name> not allowed here)

4.Complier is responsible to generate default construtor. If user not writes any constructor then only compiler generates default constructor.

Note : Compiler always check 2 things,

1. whether programmer has provided any constructor or not [If not provided any constructor, compiler will generate default constructor which has always one line inside constructor as super() like below]

<ConstructorName>(){

Super();

}

1. Whether programmer has declared constructor properly or not. i.e. while declaring constructor by programmer, if he miss to provide first line inside constructor as super(), compiler internally will add super().

Ex: Programmer’s program,

Public class Demo{

Public Demo(){

Syso(“print some thing”);

}

}

Compiler will treat it as below,

Public class Demo{

Public Demo(){

Super();

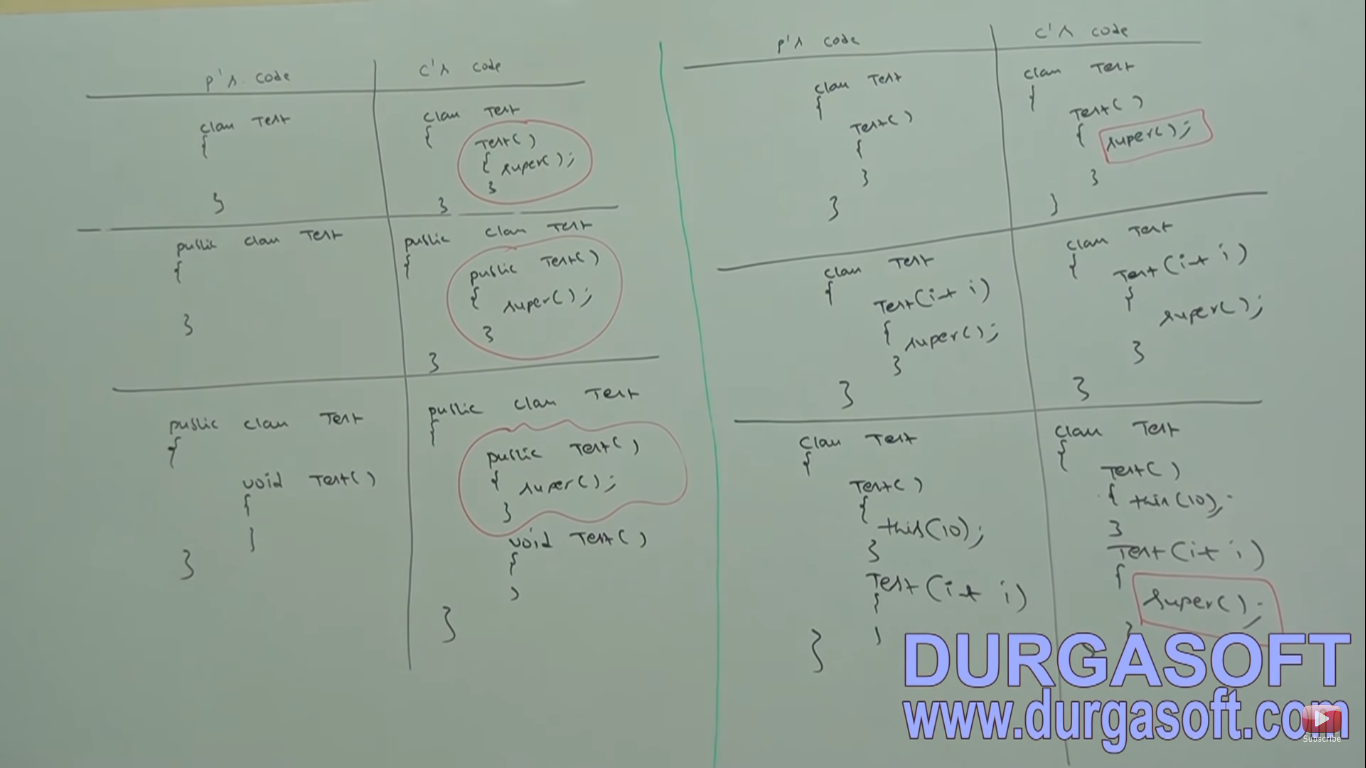
Syso(“print some thing”);

}

}

**ProtoType of default constructor :**

1. Default constructor is always no-arg constructor.
2. The access modifier of default constructor is exactly same as access modifier of Class. (This is rule is applicable only for public and default [As private and protected not applicable for high level class, same applicable to default constructor])
3. It contains only one line inside default constrctor **super().** It is no argument call to super class constructor.



Instance Block VS Constructor is different.

1. For each object creation both Instance block and constructor will execute.
2. Constructor is used for initializing object and Instance block is used for other than initializing object.
3. Instance block is executes first and then constructor execution will follow, if class having both instance block and constructor.

Ex : Interview Question :

Q) Print the number of objects created for a class?

Ans) public class Demo{

Static int count = 0;

{

Count++;

}

Demo(){

}

Demo(int i){

}

Demo(double db){

}

Public static void main(String[] args){

Demo d1= new Demo();

Demo d2 = new Demo(10);

Demo d3 = newDemo(10.5);

Syso(“number of objects created : “+count);

}

}

Explanation : As for every object creation, instance block is executes, count use to increase one time as per code.

Note :

1. If programmer declared private constructor in a class, others can’t create object of that respective class.
2. Constructor Overloading is possible ,Overriding is not applicable.
3. Super() / this() -
   1. we can use only constructors
   2. Only in first line of constructor
   3. Only one but not simultaneously. i.e. should not use both at a time.

**Difference between super(),this() and super,this keywords?**

1. **super(),this() – These are constructor calls to call super class and current class constructors**

**We can use only in constructor as first line.**

1. super,this – These are keywords to refer super class and current class instance variables.
   1. We can use anywhere except static area.