Dt : 7/9/2022

faq:

wt is the behaviour of constructor declared with return_type?

=>If the Constructor is declared with return_type then it is considered as normal instance method.

Ex : DemoCon3.java

```
class CTest3 //SubClass
{
        static CTest3()
        {
                System.out.println("====0-parameter con===");
        }
  int CTest3()
        {
          System.out.println("====Display from CTest()===");
          return(123);
        }
}
class DemoCon3 //MainClass
{
        public static void main(String[] args)
        {
                CTest3 ob = new CTest3();//Con_Call
```

```
                int k = ob.CTest3();
                System.out.println("The value k:"+k);
        }
}
```

o/p:

====0-parameter con===

====Display from CTest()===

The value k:123

==========

faq:

define static Constructor?

=>There is no concept of static constructor in Java,because constructor means executed while object creation process and which cannot be class_level_component.(Compilation Error)

==========

(b)Constructors with parameters:

=>The constructors which are declared with parameters are known as parameterized Constructors or Constructors with parameters.

=>we pass parameters to the Parameterized constructors while con_call,which means while object creation process.

Ex : DemoCon4.java

```
class CTest4 //SubClass
```

```
{
        CTest4(float k)
        {
    System.out.println("====CTest4(k)====");
                System.out.println("The value k:"+k);
        }
        CTest4(int x,int y)
        {
                System.out.println("====CTest4(x,y)====");
                System.out.println("The value x:"+x);
                System.out.println("The value y:"+y);
        }
        void dis(float z)//Instance method memory in both objects
        {
                System.out.println("====dis(z)====");
                System.out.println("The value z:"+z);
        }
}
class DemoCon4 //MainClass
{
        public static void main(String[] args)
        {
                System.out.println("*****ob1*****");
                CTest4 ob1 = new CTest4(12.34F);//Con_call
```

```
                ob1.dis(123.45F);//method_call
        System.out.println("*****ob2*****");
                CTest4 ob2 = new CTest4(11,12);//Con_Call
                ob2.dis(234.78F);//method_call
            }
}
```

o/p:

******ob1******

====CTest4(k)====

The value k:12.34

====dis(z)====

The value z:123.45

******ob2******

====CTest4(x,y)====

The value x:11

The value y:12

====dis(z)====

The value z:234.78

==========

**imp*

Multiple Constructors in a Class:

=>we can declare Multiple Constructors in a class,but

the constructors are executed based on con_call available in

Object creation process.

```
==================================================
*imp
Advantage of Constructors:
 =>Constructors are used to initialize instance variables
while object creation process,and which saves the execution
time and generate HighPerformance of an application.
 ==================================================
Ex_program : DemonCon5.java
import java.util.Scanner;
class User //SubClass
{
        //Instance Variables memory in Object
        String name,mailId;
        long phNo;

  //local Variables memory in method
        User(String name,String mailId,long phNo)
        {
            this.name = name;
            this.mailId = mailId;
                this.phNo = phNo;
        }

        void getUser() //Getter method
```

```
        {
                System.out.println("===User details====");
                System.out.println("User Name:"+name);
                System.out.println("User MailId:"+mailId);
                System.out.println("User PhNO:"+phNo);
        }
}
class DemoCon5 //MainClass
{
        public static void main(String[] args)
        {
                Scanner s = new Scanner(System.in);
                System.out.println("Enter the UserName:");
                String uName = s.nextLine();
      System.out.println("Enter the MailId:");
                String mId = s.nextLine();
      System.out.println("Enter the PhoneNo:");
                long phNo = s.nextLong();

                User u = new User(uName,mId,phNo);//Con_call
      u.getUser();
        }
}
```

o/p:

Enter the UserName:

nit.v

Enter the MailId:

v@gmail.com

Enter the PhoneNo:

7878781234

===User details====

User Name:nit.v

User MailId:v@gmail.com

User PhNO:7878781234

==========

faq:

define 'this' keyword?

=>"this" is a pre-defined Non-Primitive datatype variable

holding object_reference of current running class.

Note:

=>when we want to load data from local variables to

instance variables and if they are having same names then we

use 'this' keyword.

==========