Taking user Input through Scanner class

*) Scanner class is a way of taking user input

*) Scanner class is inside the package of java.util

e.g.

```
import java.util.Scanner;
class A
{
    p. s. v. m(String args[])
    {
        Scanner s = new Scanner(System.in);
```

creating object   for i/p

```
        s.o.p("Enter name");
        String name = s.next();
```

→ method of Scanner class

```
        s.o.p("Enter gender");
        char ~~String~~ gender = s.next().charAt(0);
        s.o.p("enter Age");
        int age = s.nextInt();
        s.o.p("Enter Ph.no");
        long Phno = s.nextlong();
```

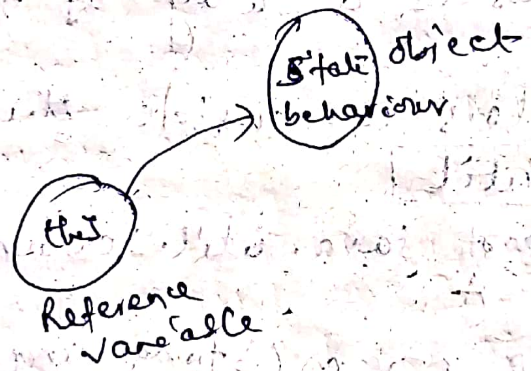
~~Static~~ ~~Method~~

~~Static~~

```
[ S.O.P("Enter salary");
[ double d = ~~in~~. s.nextDouble();
```

This Keyword

*) This key word is a Reference variable that Refers to the current object.



*) 'this' keyword can be used to Refer current class instance variable.

e.g.

```
class xyz
{
   int i;
   void show( int i)
   {
      this. i = i;
   }  S.o.p(i);
   p.s.v.m(      )
   { xyz ob = new xyz()
      {
         ob. show(10);
      }
   }
}
```

instance variable

local variable

*) This keyword can be used to refer current class instance Variable.

*) This keyword can be used to invoke current class method (implicitly).

*) This keyword can be used to invoke current class constructor.

*) It can be used to pass as an argument in the method call.

*) used to pass as an argument in the construct or call

Current class method

```
class test
 {
  ~~int i~~;
  void display()
   {
    sop("Hello");
   }
  void show()
   {
    display();
   }
  psvm()
   {
    test obj = new test();
    obj.show();
   }
 }
```

⟹ Implicitly compiler will add ~~this~~ keyword to ~~the~~ call ~~the~~ instance method i.e. this.display();

Current class constructor

```
class test
{
   test()
    { sop("no arguments");
    }  this(10);
   test(int a)
    { sop("parameterized");
    } this();
```

```
psvm()
{
 test obj = new test(10);
}
```

call the ~~ew~~ default constructor.

argument in the method call

```
class test
{  void m1(test ob)
   {
       S.o.p("I am in M1");

   }
   void m2()
   {   m1(this);
   }
  p s v m ( )
  { test  t = new test();
    t.m2();
  }
```

As an Argument.

used to pass as an argument in the constructor call.

```
class test
{   test(test ob)
    {

    }

    :
}
  p s v m()
  {
  test t = new test (this);
```