

Lec - 1

Var - args Method

* Main reason behind the varargs origin.

(1) What will happen if you don't know how many arguments you need to pass to a method?

(2) What will happen if you want to pass unlimited variables to a method?

There is a simple answer for this question in java. It is simple, and there is varargs. Using this concept you can send as many as parameters of the same type without any restrictions.

Syntax of Varargs :-

return_type method_name (data_type ... Variable_name)

Example :-

void sum (int... x)

{

}

Lec - 2

Class A

```
§ void sum (int... a)
§ System.out.println ("abhi ");
§ }
```

Class Demo3

```
§ psum (String arr [] )
§ }
```

```
A a = new A ();
a.sum (10, 20);
a.sum (10, 20, 30);
a.sum (10, 15, 2, 8); }
```

abhi
abhi
abhi
abhi

Class A

```
§ void sum (int... a)
§ s. o. p (a.length); }
```

Class Demo

```
§ psum (String arr [])
§
A a = new A ();
a.sum ();
a.sum (10);
a.sum (10, 20, 30); }
```

0
1
3

Lec-73

Display Data From Vectors

For Loop

Class A

```
§ void sum (int... a)
§
for (int i=0; i<a.length; i++)
§
    s.o.p(a[i] + "\t");
§
s.o.println();
```

10
10 20 30
1 2 3 4 5

class Demo

```
§ psum (String arr[])
§
A a = new A();
a.sum (10);
a.sum (10, 20, 30);
a.sum (1, 2, 3, 4, 5);
```

For Each Loop

void sum (int... a)

```
§
for (int i:a)
§
    s.o.p (i + "\t");
§
s.o.println();
```

10
10 20 30
1 2 3 4 5

Performing Operation :-

class sum

```
§ void sum (int... a)
§
int s=0;
for (int i:a)
§
    s = s+i;
```

```
Sum = 10
Sum = 60
Sum = 23
```

```
§ s.o.println ("sum=" + s);
§
§
class Demo
```

§ psum (String arr[])
§

```
A a = new A();
a.sum (10);
a.sum (10, 20, 30);
a.sum (10, 1, 12);
```

Lec-7

Date _____

(1) Agar name Vectors help to handle simple geometric operations like dot product, cross product, vector addition, subtraction, etc.

Vectors Ko last mre calculate hoyi

vector ayegi.

Class A

void elm (int a, int b)

int s = 0;

for (int i = a;

s += i;

sum += s;

Class Demo

sum (string arr)

A a = new AC;

a.sum (10, 20, 30);

a.sum (11, 10, 20, 30);

void sum (int ... a)

expected

void sum (int ... a)

public static void main (String args[])

int a = 10, b = 20, c = 30;

System.out.println (a + " " + b + " " + c);

Output :- 10 20 30

Benefit / Advantage of Vectors

(1) Agar name KTS method ma otharay type ka parameter use karna hai to hamesha banana pdgega. Pd.
parameter value bihi to use execute krene
banana value helps to calculate array type ka
se ham kinti thi value cao q
parameter pass kta direktly.

(2) Agar name KTS method ma otharay type ka
parameter value bihi to use execute krene
banana value helps to calculate array type ka
se ham kinti thi value cao q
parameter pass kta direktly.

Lec -T5

DATE
PAGE

There can be only one vari-args parameter in the method. we can't use more than one vari-args variargs in a single method.

class A

```
§ void sum (int... a)  
{  
    System.out.println ("Softwaves");  
}
```

```
§ void sum (int[] a)  
{  
    System.out.println ("Bhanu");  
}
```

class Demo

```
§ psvm (String arr[])
{  
    A a = new A();  
    a.sum (10, 20);  
}
```

Error :- Can not declare both sum (int[])
and sum (int...) in A

Hm ek hi class me Variargs and Array type ke parameter pass nhii kta skte hain.

Lec -T6

DATE
PAGE

Method Overloading is possible in vari-args.

class A

```
§ void sum (int... a)  
§ System.out.println ("Bhanu");  
§
```

```
§ void sum (String... a)  
§ System.out.println ("Shailu");  
§
```

```
§ void sum (boolean... a)  
§ System.out.println ("momo");  
§
```

Shailu
momo

Class Demo

```
§ psvm (String arr[])
{  
    A a = new A();  
    a.sum (10, 20);  
}
```

Bhanu
Momo
shailu

```
a. (true, false, true);  
a. ("sum", "situ");  
§  
§
```

If we pass or call method sum like
a.sum (); then ambiguity error will occur.
Because the compiler gets confused to call which sum method.

Lec-7

DATE
PAGE

Vari-args hameshai last me execute hoga
hai

class A

```

{ void sum (int... a)
{ S.o.p ("Bhanu");
}
void sum (int a)
{ S.o.p ("Shailu");
}
}
```

Class Demo

```

{ psum (String arr[])
{ A a = new A();
a.sum();
a.sum(10);
a.sum (10, 20);
}
}
```

Vari-args badal me execute hoga agr
same method hai with parameter
tak.

Bhanu
Shailu
Bhanu

Lec-8

DATE
PAGE

We can also pass an array in
var-args.

Variable argument

void sum (int... a)
{
sum ();
sum (10);
sum (10, 20);
int x[] = {10, 20, 30, 40};
sum (x);

(All can run)

(only last will run)

Lec-9

To pass 2D Array in Vari-args

```

void sum (int[][]... a)
{
    for (int i[] : a)
        for (int j : i)
            S.o.p (j + " ");
    S.o.pln ();
    S.o.pln ();
}
```

Passing 3D Array in Var-args

```
void sum (int [][]... a)
{
    for (int j[] [] : a)
    {
        for (int j[] : i)
        {
            for (int k : j)
            {
                System.out.print (" " + k);
            }
            System.out.println ();
        }
        System.out.println ();
    }
}
```

Lec - 10

Passing 2 Arrays in Var-args at a single time.

```
Class A
{
    void sum (int []... a)
    {
        System.out.println ("Shailu!");
    }
}

Class Demo
```

```
psvm (String arr [] )
```

```
{
    int x [] = {10, 20, 30};
    int y [] = {1, 2, 35};
    A a = new A ();
    a.sum (x, y);
}
```

Shailu

Different syntax of Var-args

- (1) void sum (int ... a)
- (2) void sum (int ... , a)
- (3) void sum (int ..a a)
- (4) void sum (int ... , a)

→ Can be run

But

```
void sum (int ..a a)
void sum (int a ..a)
```

Not run

We can also use main method as
public static void main (String [] arr)