

23

Friday

# Working with Command Line Arguments

in <sup>4th week</sup> java.

January

Let us start with a small programs -

class Demo

```

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

```

```

        System.out.println("Arguments passed" +

```

```

            args[0]);

```

```

    }

```

```

}

```

Compile :- javac Demo.java ←

java Demo ←

This statement will show an exception error "Array Index Out of Bounds" because here we have not passed any argument while running the program with java.

That means we have to pass atleast one argument while running the program.

That means : java Demo Hello ←

The output :- Arguments passed Hello

→ Hello is stored in args[0] and gets printed.

Now, let us try, get two numbers from Command prompt and get their sum :-

Arguments are always treated as string.

JANUARY							FEBRUARY						
S	M	T	W	T	F	S	S	M	T	W	T	F	S
			1	2	3		1	2	3	4	5	6	7
4	5	6	7	8	9	10	8	9	10	11	12	13	14
11	12	13	14	15	16	17	15	16	17	18	19	20	21
18	19	20	21	22	23	24	22	23	24	25	26	27	28
25	26	27	28	29	30	31							

4th week  
024-341

24  
Saturday

Hence, if we have to treat as numbers for ~~calculating~~ performing mathematical operations, then we need to convert it to integer first then we can do it.

For that, we need to take a variable of int type then typecast string to integer and store it as:-

```
int i;
i = Integer.parseInt(args[0]);
```

Here parseInt() method belongs to Integer class, which is static method.

Now, Program becomes:-

25 Sunday

```
class Prog2
{
    public static void main(String a[])
    {
        int i, j, sum=0;
        i = Integer.parseInt(a[0]);
        j = Integer.parseInt(a[1]);
        sum = i + j;
        System.out.println("Sum of i and j is: " + sum);
    }
}
```

just a variable name

26

5th week  
026-339

Now compile and run it as follows:-

Open Comm  
January 0

Monday

C:\Java> javac Prog2.java ←

C:\Java> java Prog2 2 5 ←

Output will be :- Sum of i and j is 7.

10

## Types of Variables in Java

11

There are two types of variables used in Java — primitive variables — used to represent primitive values such as `int i=50;`

12

13

14

15

Reference Variables — used to refer

16

objects such as:- `Prog2 a1 = new Prog2();`

17

Here a1 is a reference variable of class `Prog2`.

18

19

Based on the purpose and <sup>position of</sup> declaration of all variables can be divided into three types:-

20

21

Notes

Instance Variable

Static Variable

Local Variable

Value of variable is varied from object to object.

Value is not varied from object to object.

Created inside a block.

JANUARY						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

FEBRUARY						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

### Instance Variable

cannot be accessed from static area directly.

we have to use object name to call instance variable

### Static Variable

Static variables are also known as class level variables.

All objects of the class can have the same value as given to static variable.

### Local Variable

It can be created inside a constructor or method.

Initialization is necessary before using it.

### Instance Variable: -

- Scope is similar with scope of object.
- Should be declared within the class directly, but outside of any method, block or constructor.

ex:-

```

class Test-
{
    int i=100;
    public static void main(String args[])
    {
        Test t1 = new Test();
        System.out.print(t1.i);
    }
}

```

Initialization of instance variable is not mandatory.

Wednesday

Variable:  $i$   $\begin{cases} \rightarrow a1.i = 200; \\ \rightarrow a2.i = 400; \end{cases}$

Suppose in above program we have given statement as:-

```
Test a1 = new Test();
```

```
a1.i = 200;
```

```
Test a2 = new Test();
```

```
a2.i = 400;
```

Static Variable :- When we have to use a common value for a variable which is constant for all objects of the class, then we need to use static variable.

It is also declared directly inside the class outside of any block, method or constructor with the keyword static.

eg:-

```
class Test
{ static String College = "SSC"; //static
  int stdid; //instance
```

```
public static void main(String args[])
{
  Test t1 = new Test();
```

Notes

JANUARY						
S	M	T	W	T	F	S
			1	2	3	
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

FEBRUARY						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

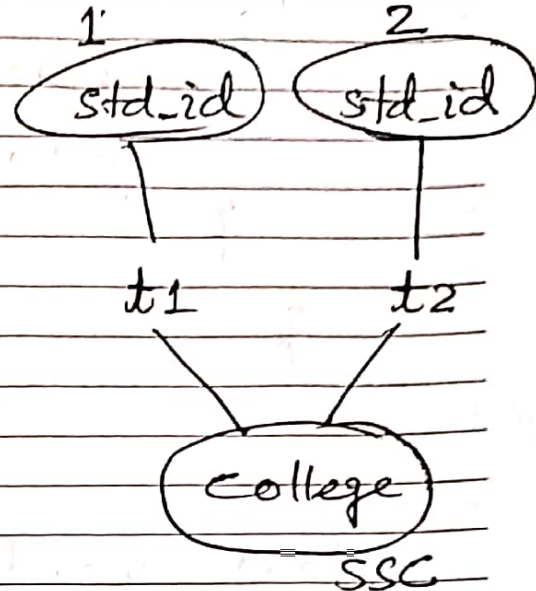
```
Test t2 = new Test();
```

```
t1.std_id = 1;
```

```
t2.std_id = 2;
```

```
}
```

```
}
```



→ static variable's scope is same as scope of class.

→ Need not be initialized (not mandatory) JVM will automatically initialize if not done.

→ declared inside the class outside any block or method.

→ Static variables are also called class level variables or fields.

### Local Variables :-

They are declared inside any block, method or constructor and available locally. They need to be initialized mandatory.

```

class vardemo {
    int i;
    public static void main(String args[])
    {
        vardemo obj = new vardemo();
        System.out.println(obj.i);
    }
}

```

javac vardemo.java  
 java vardemo  
Result will be: 0

```

class vardemo {
    int i = 10;
    public static void main(String args[])
    {
        vardemo obj = new vardemo();
        vardemo obj1 = new vardemo();
        obj.i = 150;
        obj1.i = 500;
        System.out.println(obj.i);
        System.out.println(obj1.i);
    }
}

```

Output :-  
 150  
 500

JANUARY						
S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

FEBRUARY						
S	M	T	W	T	F	S
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28

5th week  
011 134

31  
Saturday

```
class vardemo {
```

```
    int i = 10;
```

```
    static String name = "SSC";
```

```
    public static void main (String args[]) {
```

```
    {
```

```
        vardemo obj = new vardemo();
```

```
        System.out.println(name);
```

```
    }
```

```
}
```

Output: SSC

Static variables can be used for counting object creation of a class as:

01 Sunday

```
class vardemo
```

```
{
```

```
    static int i = 0;
```

```
    vardemo() { i++;
```

```
        System.out.println(i);
```

```
    }
```

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

```
{
```

```
    vardemo v1 = new vardemo();
```

```
    vardemo v2 = new vardemo();
```

```
    vardemo v3 = new vardemo();
```

```
} }
```