

## Types of variable

- a) Local
- b) Instance/Global
- c) Static

**a) Local variable** :- A variable which declare inside method/block/constructor called as local variable

Scope of local variable remains only within the method/block/constructor.

Local variable is also called temporary variable.

### Program :

```
package TypesofVariable;
```

```
public class Prog1 {
```

```
    static int d = 100; //Global / Instance variable
```

```
    public void add()
```

```
    {
```

```
        int a = 10, b = 20, c = 0; // Local
```

```
        c = a + b;
```

```
        System.out.println("Add = " + c);
```

```
        System.out.println("D in add method = " + d);
```

```
    }
```

```
    public void multiply()
```

```
    {
```

```
        System.out.println("D = " + d);
```

```
    }
```

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

```
        Prog1 z = new Prog1();
```

```
        z.add();
```

```
//          System.out.println("A = " + a); We cant access local variable outside the
method
```

```
        System.out.println("D in main method = " + d);
```

```
    }
```

```
}
```

**b) Instance/Global variable**:-A variable which declare inside the class but outside of all method/block/constructor is called instance variable

scope of global variable remains thought the class.

global variable is also called permanent variable.

**c) Static/ Class variable** :- A variable which declare with the help of static keyword is called as static variable.

1. static variable call from same class --> variableName

2. static variable call from diff class--> className.variableName

Note: we can access static global variable in both static & non-static method

### Program:

```
package TypesofVariable;
```

```
public class Prog2 {
```

```
    int num1 = 10;          // 4 x no of object (3) = 12          //Instance / Global /
```

```
Non static
```

```
    static int num2 = 10; //4 byte          //Instance / Global / static
```

```
    public void increment()
```

```
    {
```

```
        num1++; //11
```

```
        num2++; //10 -> 11 - 12 - 13
```

```
        System.out.println(num1 + " " + num2);
```

```
    }
```

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

```
        Prog2 x = new Prog2(); //Num1 = 10 num2=10
        x.increment();
```

```
        Prog2 y = new Prog2(); //Num1 = 10 num2=11
        y.increment();
```

```
        Prog2 z = new Prog2(); //Num1 = 10 num2=12
        z.increment();
```

```
        Prog2 z11 = new Prog2(); //Num1 = 10 num2=13
        z11.increment();
```

```
    }
```

```
}
```

```
//Prog2 class -> num2=10
```

```
//x-> num1=10 y-> num=10 z->num=10
```

### This Keyword

Every class have unique reference number, so reference variable (object) refer the unique reference number of class also this keyword refer the unique reference number.

- This keyword refer to current object inside the method or constructor
- Whenever the name of instance variable and local variable are same then JVM confused which one is local variable and which is instance variable so to avoid this we should used this keyword
- It is also used when we want to call zero parameter constructor of its own class
- It is also used to call parameterized constructor of its own class

### Program :

```
package ThisProg;
```

```

public class Prog1 {

    public void show()
    {
        System.out.println(this);
    }

    public static void main(String[] args) {
        Prog1 x = new Prog1();
        x.show();
        System.out.println(x);
        System.out.println("*****");
        Prog1 y = new Prog1();
        y.show();
        System.out.println(y);
    }

}

```

**Program:**

```

package ThisProg;

```

```

public class Prog2 {

    int num1 = 10;

    public void display()
    {
        int num1 = 20;
        System.out.println(num1); //20
        System.out.println(this.num1);
    }

    public static void main(String[] args) {

        Prog2 z = new Prog2();
        z.display();
    }

}

```