SYNTAX
TECHNOLOGIES

JAVA

Class 18

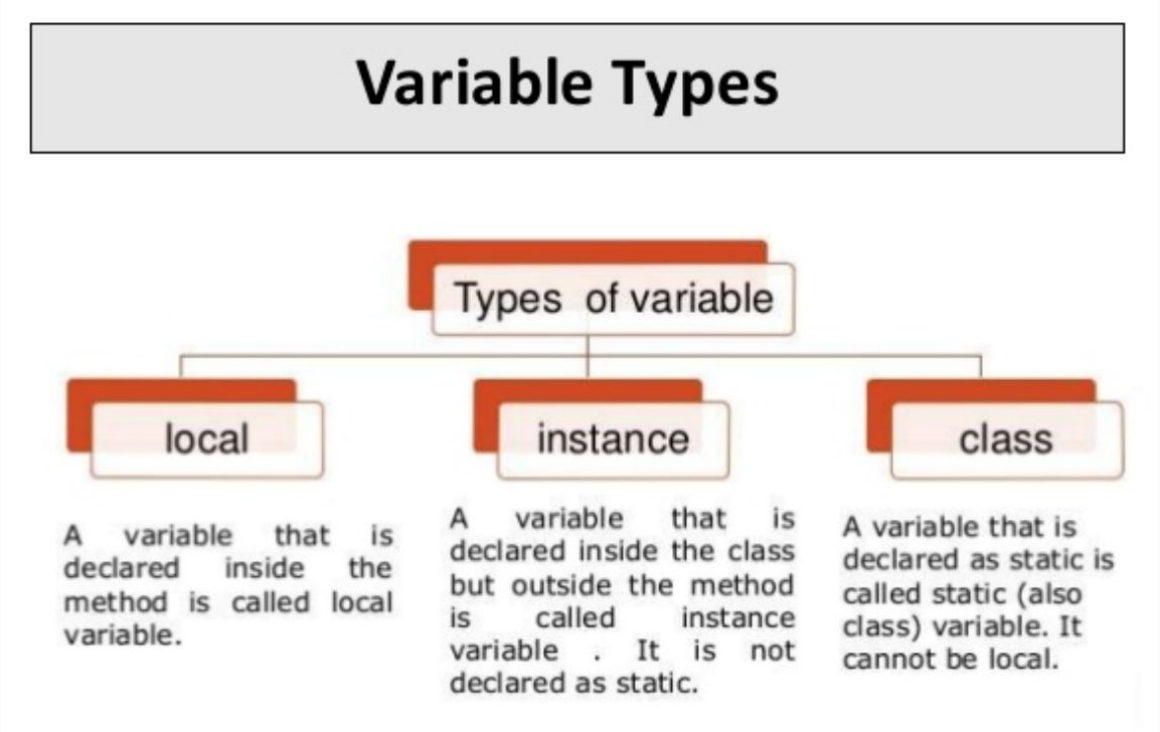
Agenda

Variable types in JAVA:
- Local
- Instance
- Static

Static Keyword

Types of variables in Java

1. Local variables.
2. Instance variables.
3. Static variables.



Local Variable

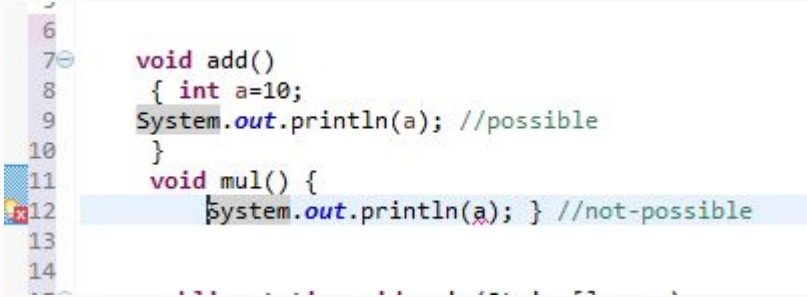
The variables which are declare inside a method or constructor or blocks those variables are called local variables.

```
    public static void main(String[] args)
        { //local variables
        int a=10;
        int b=20;
        System.out.println(a+b);

}
}
```

Local Variable

It is possible to access local variables only inside the method or constructor or blocks only, it is not possible to access outside of method or constructor or blocks.
Local variables memory allocated when method starts & memory released when method completed



Local Variable

- Local variables are declared in methods, constructors, or blocks.
- Local variables are created when the method, constructor or block is entered and the variable will be destroyed once it exits the method, constructor, or block.
- Access modifiers cannot be used for local variables.
- A variable declared inside the body of the method is called local variable.
- You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.
- A local variable cannot be defined with "static" keyword.

```
public class VariableExample {
        public String myVar="instance variable";   // instance variable
        public void myMethod(){
                // local variable
                String myVar = "Inside Method";
                System.out.println(myVar);
        }
   public static void main(String args[]){
          VariableExample obj = new VariableExample();   // Creating object

      /* We are calling the method, that changes the  value of myVar. We are displaying myVar
again after the method call, to demonstrate that the local  variable scope is limited to the
method itself.
       */
      System.out.println("Calling Method");
      obj.myMethod();
      System.out.println(obj.myVar);
   }
}
```

```
Calling Method
Inside Method
instance variable
```

Instance Variable

The methods and variables defined within a class are called members of the class.

A variable which is declared inside the class but outside the method is called instance variable. It is not declared as static.

Why the name is instance: Variables defined within a class are called instance variables because each instance of the class (that is, each object of the class) contains its own copy of these variables.

Thus, the data for one object is separate and unique from the data for another.

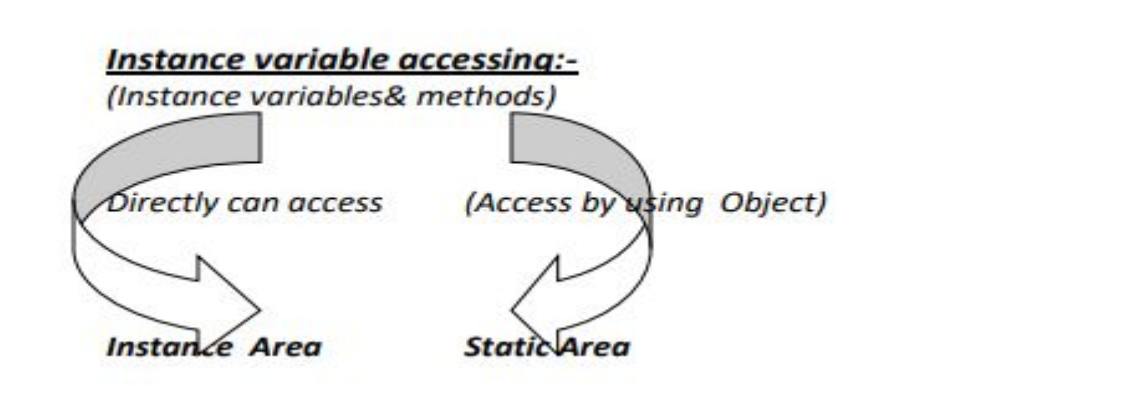
It is called instance variable because its value is instance (object) specific and is not shared among instances.

Important Points about Instance Variable

Instance variables are used in a class, but outside a method, constructor or any block.

A slot for each instance variable value is created, when space is allocated for an object in the heap.

Instance variables can be declared in the class level before or after use.



```
public class InstanceVarExample {
  String myInstanceVar="instance variable";

  public static void main(String args[]){
      InstanceVarExample obj = new InstanceVarExample();
      InstanceVarExample obj2 = new InstanceVarExample();

      System.out.println(obj.myInstanceVar);
      System.out.println(obj2.myInstanceVar);

      obj2.myInstanceVar = "Changed Text";

      System.out.println(obj.myInstanceVar);
      System.out.println(obj2.myInstanceVar);
  }
}
```

```
instance variable
instance variable
Changed Text
instance variable
```

```
public class InstanceVariable{

    public String name;// this instance variable

    public InstanceVariable(String consName) {
     name = consName;
     System.out.println("Your Name is: " + name);
     }

    public static void main(String args[]) {
        InstanceVariable r = new InstanceVariable("atnyla");
    }
}
```

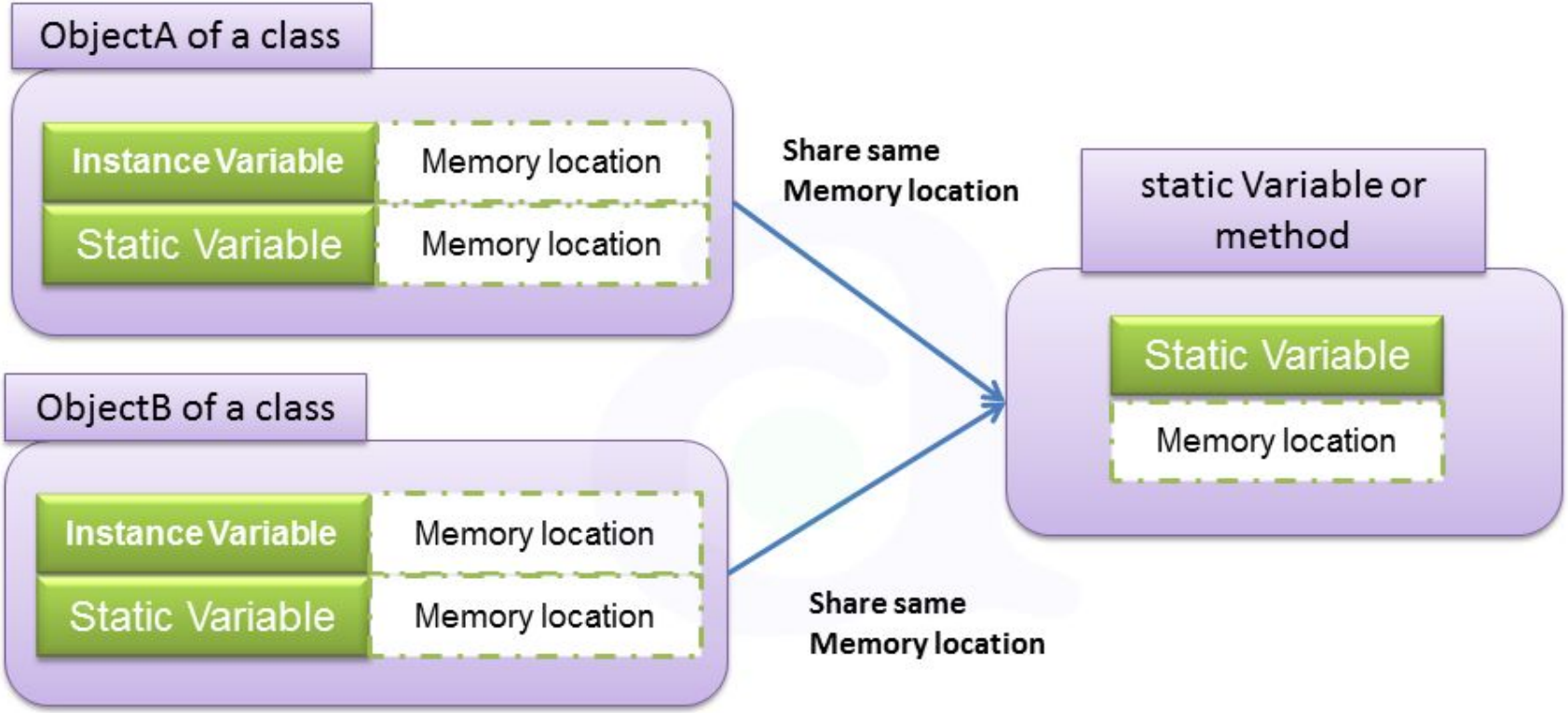
```
Your Name is: atnyla
Press any key to continue . . .
```

Static variable

- A variable which is declared as static is called static variable.
- Static variables are declared with the static keyword in a class, but outside a method, constructor or a block.
- It cannot be local.
- You can create a single copy of static variable and share among all the instances of the class.
- Memory allocation for static variable happens only once when the class is loaded in the memory.

Static variable

- Static variables are also known as class variable because they are associated with the class and common for all the instances of class.
- Static variables store values for the variables in a common memory location.
- Because of this common location, if one object changes the value of a static variable, all objects of the same class are affected.
- Use the static variable for the property that is common to all objects. For example, in class Student, all students shares the same college name.
- Use static methods for changing static variables.



static variables store values for the variables and **share a common memory location** for all objects

```
package studytonight;
class Student{
    int a;
    static int id = 35;
    void change(){
        System.out.println(id);
    }
}
public class StudyTonight {
    public static void main(String[] args) {

        Student o1 = new Student();
        Student o2 = new Student();
        o1.change();
        Student.id = 1;
        o2.change();
    }
}
```

```
35
1
```

Static keyword

- static is a non-access modifier in Java.
- static keyword can be used with variables, methods, blocks and nested class.
- static keyword belongs to the class than an instance of the class.
- static methods can access all static variables and other static methods.
- static methods cannot access instance(non static) variables and methods directly; they need some object reference to do so