

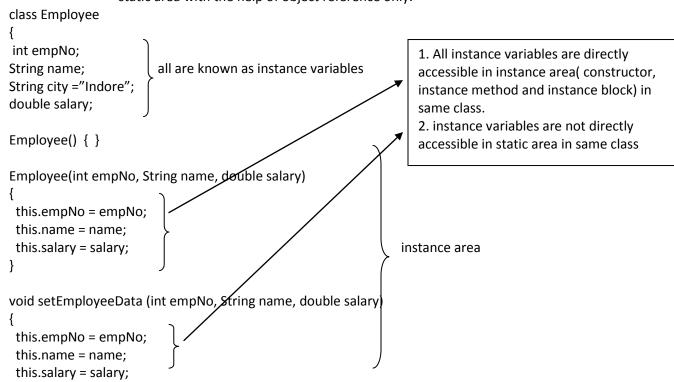
Types of Variables

Based on the position of declaration and behavior we can categorize variables in three types:

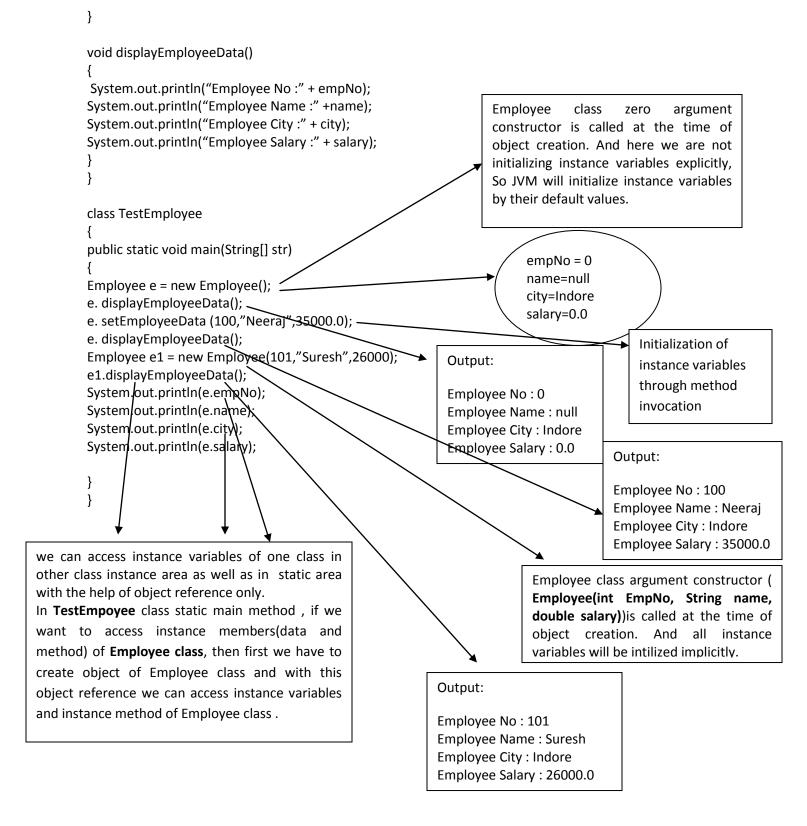
- 1. instance variable
- 2. static variable
- 3. local variable

Instance Variables:

- **1.** Instance variables are those variables declared inside a class but outside of any method, constructor or block and without static modifier.
- 2. These variables are created when an object of the class is created and destroyed when the object is destroyed.
- 3. For each and every object a separate copy of instance variables is created.
- 4. instance variables value may or may not vary from object to object.
- 5. All objects are stored in heap memory and as a part of object all instance variables are also stored on heap memory.
- 6. Accessing of instance variables
 - **In same class I.** All instance variables are directly accessible in instance area(constructor, instance method and instance block)
 - II. instance variables are not directly accessible in static area in same class
 - **In Other class** we can access instance variables of one class in other class instance area as well as static area with the help of object reference only.









Static Variables:

- 1.static variables are those variables declared inside a class but outside of any method, constructor or block and with static modifier.
- 2. These variables are created when a class is loaded into the memory and destroyed when the class is unloaded from the memory.
- 3. only one copy of static variables is created in memory at the time of class loading and this copy is shared to all the objects.
- 4. All static variables are stored in method area.
- 5. If we are not initializing static variables explicitly, then it is the responsibility of JVM to initialize static variables by their default values .
- 6. Accessing of static variables
 - **In same class** All static variables are directly accessible in instance area(constructor, instance method and instance block) as well as in static area
 - **In other class** we can access static variables of one class in other class instance area as well as in static area with the help of object reference only.

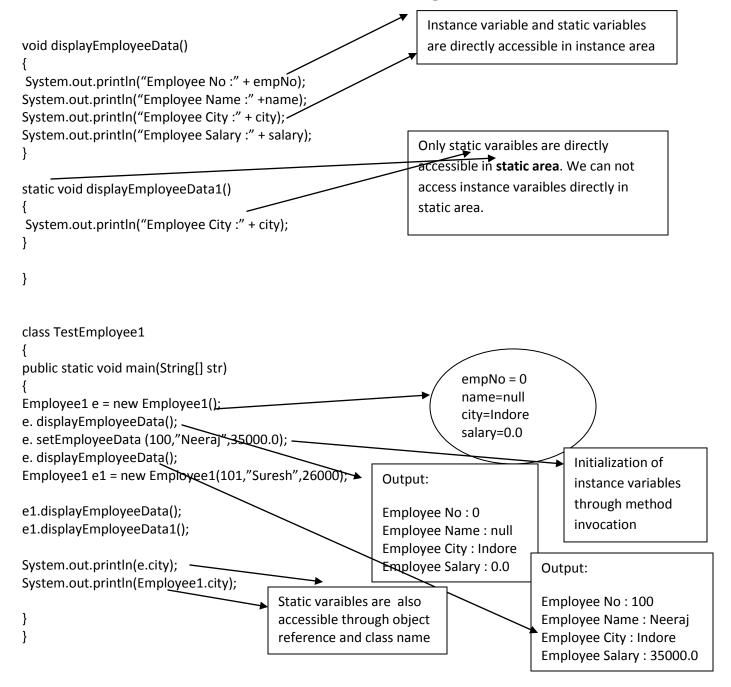
We can also access static variables with the class name also

So static variables can be accessible in three ways -

- 1. Direct
- 2. With object reference
- 3. With class Name

```
class Employee1
int empNo;
String name;
static String city ="Indore";
                               // static variable
double salary;
Employee1() { }
Employee1(int empNo, String name, double salary)
this.empNo = empNo;
this.name = name;
this.salary = salary;
void setEmployeeData (int empNo, String name, double salary)
this.empNo = empNo;
this.name = name;
this.salary = salary;
}
```





Local variable

- 1.local variables are those variables declared inside a method, constructor or block.
- 2. These variables are created when a method is invoked or constructor/ block is executed and destroyed at the completetion of execution of block, constructor and method.
- 3. All local variables are stored in stack area.



4. If we are not initializing local variables explicitly, then at the time of accessing we will get complilation error. It is the responsibility of programmer to initialize all local variables.

Important Not Regarding instance, static and local varaibles:

- 1. For instance variables, access modifiers (no-midifier/default, public, private and protected) can be given. Also we can use non-access modifier final ,volatile and transiant keyword with instance varible.
- 2. For static variables, access modifiers (no-midifier/default, public, private and protected) can be given. Also we can use non-access modifier final ,volatile with static variable, but we can not use transiant keyword with static variable
- 3. For local variables, access modifiers (no-midifier/default, public, private and protected) can not be given. We can use only use non-access modifier final with local varible. Other non-access modifiers are not allowed with local variables
- 4. synchronized, abstract, native, strictfp non-access modifier are not allowed with instance, static and local variables.