Working with static and non-static

Static members

Non-static members

|  |  |
| --- | --- |
| Program | Execution |
| class Apple{  static int greenApple=3;  int redApple;  public static void main(String ar[]){    }  static void eatGreenApple(){  }  void eatRedApple(){  }  }  Apple a1 = new Apple();  new Apple(); | greenApple=3  eatGreenApple()  main()  Apple class  redApple  eatRedApple()  redApple  eatRedApple() |

You can declare the variables and methods in a class with the key word **static.** Then these members will become **static members.**

Variables and methods declared in class without static keyword is called as **non-static members or instance members or object members.**

Static members belong to **class** & non static/Instance members belong to **Object.**

The static keyword in Java is mainly used for memory management. The static keyword in Java is used to share the same variable or method of a given class.

Static members are only one per class (one copy). Whereas instance members, each object **has its own copy of non-static** members/instance members.

Static variable also called as class variables.

Non static variables also called as instance variables.

Example:

class VariableDemo{

static int a=10; //static variable/class variable (Fields)

int b=20; //non-static variable/instance variable

public static void main(String args[]){

int c=30; //local variable

}

}

* Whenever we use class variable (static) or method (static) always use class name.memberName. Don’t use it directly.

|  |  |
| --- | --- |
| Using static variables and methods | Using non-static variables and methods |
| class Demo{  static int i=20;  public static void main(String a[])  {  Demo.test();  }  static void test(){  System.out.println(Demo.i);  }  } | class Demo{  int a=20;  public static void main(String a[])  {  Demo d1=new Demo();  d1.a=30;  d1.test();  }  void test(){  System.out.println(a);  }  } |

**Note:**

* Static members can be accessed anywhere in the program. It can be called from static and non-static method **directly or with class\_Name.member\_Name.**
* Non-static members can be accessed **by non-static methods** **directly**. Non-static members from a static method we have to use reference\_Variabe**.**(dot) OR object**.**(dot) member\_Name.
* It is possible to use non static members inside non static methods directly
* **To access non static members inside static method we need to create object**
* **public** **class** StaticDemo {
* **static** **int** *a*=10;
* **int** b;
* **public** **static** **void** main(String[] args) {
* // **TODO** Auto-generated method stub
* //StaticDemo s1= new StaticDemo();
* System.***out***.println(*a*);//**using static member wihout using class name**
* *StaticDemo1*();
* }
* **static** **void** StaticDemo1()
* {
* *a*=20;//using static member wihout using class name , using directly
* System.***out***.println(*a*);
* StaticDemo s1= **new** StaticDemo();
* System.***out***.println("before value " +s1.b);// non static members can be used only through reference variable
* s1.b=30;
* System.***out***.println("after value "+s1.b);
* }
* }

In class contains two types of members

1. Static
2. Non-static

|  |  |
| --- | --- |
| Static | Non-static |
| Static members are created using static keyword on variable and method. | Non-static members are created without using static keyword on variable and method. |
| Static member belongs to class. | Non-static member belongs to object. |
| Static members are one per class. | Each object has its own copy of non-static members. |
| All static members can be accessed by using . (dot) operator on the class name. | Non-static members can be accessed by using reference variable and **.** (dot) operator. |
| Static variables are initialized when the class is loaded. | Non-static variables **are initialized when the respective object is created.** |
| Static members also called class members. | Non-static members are also called Reference members, Instance members, Object Members. |