**Java static keyword**

The static keyword in java is used to for memory management mainly , we can apply java static keyword with variables , methods , blocks and nested class . The static keyword belongs to the class then instance of the class .

* Java supports definition of global methods and variables that can be accessed without creating objects of a class . Such a members are called as static members .
* Define a variable by marking with the static methods .
* This features is useful when we want to create n variable common to all instances of a classes .
* Note : Java creates only one copy for a static variable which can be used even if the class is never instantiated .

**The static can be :**

* Variables (also known as class variable)
* Method (also known as class method)
* Block
* Nested class

If you declare any variable as a static , it is known as static variable .

* The static variable can be used to refer the common property of all objects(that is not unique for each object ) e.g . company name of employees , college name of students etc .
* The static variable gets memory only once in class area at the time of class loading .

**Advantages of static variables :**

* It makes your program memory efficient (i.e . it saves memory )

Suppose there are 2000 students in our university now all instances data members will get memory each time when object is created . All student have its unique rollNo , and name . so instance data member is good . Here university refers to the common property of all objects . If we make it static .This field will get memory only once .

**Example of static variables :**

class student {

int rollno ;

String name ;

static String college = “SPPU”

student(int r , string s){

rollNo = r ;

name = n ;

}

void display(){

System.out.println(rollNo + “ “ +name+ “ ”+college);

}

public static void main(String args[]){

student s1 = new student(37 ,”Rutvik”);

student s2 = new student(40 , “Apurva”);

s1.display();

s2.display();

}

}

Q) Why method overloading in java is not possible by changing the return type of method is changed ?

Ans – In java , method overloading is not possible by changing the return type of the method because there may arise some ambiguity .

Lets see how ambiguity may occur .