Today’s Topic:

1. Static

Static:  **Static keyword** in [Java](https://www.javatpoint.com/java-tutorial) is used for memory management mainly. We can apply static keyword with [variables](https://www.javatpoint.com/java-variables), methods.

* The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.

Why it is used?

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

**class** Student {

**int** rollno;//instance variable

String name;

**static** String *college* ="Daffodil";//static variable

//constructor

Student(**int** r, String n){

rollno = r;

name = n;

}

//method to display the values

**void** display (){System.***out***.println(rollno+" "+name+" "+*college*);}

}

//Test class to show the values of objects

**public** **class** StaticTest{

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

Student s1 = **new** Student(111,"Karan");

Student s2 = **new** Student(222,"Aryan");

//we can change the college of all objects by the single line of code

//Student.college="BBDIT";

s1.display();

s2.display();

} }

Java static method

If you apply static keyword with any method, it is known as static method.

* A static method belongs to the class rather than the object of a class.
* A static method can be invoked without the need for creating an instance of a class.
* **class** Calculate{
* **static** **int** cube(**int** x){
* **return** x\*x\*x;
* }
* **public** **static** **void** main(String args[]){
* **int** result=Calculate.*cube*(5);
* System.***out***.println(result);
* }
* }

**Method**

Method: A **method** is a block of code or collection of statements or a set of code grouped together to perform a certain task or operation. It is used to achieve the **reusability** of code. We write a method once and use it many times. We do not require to write code again and again. It also provides the **easy modification**

**Types of Java methods**

**Depending on whether a method is defined by the user, or available in the standard library, there are two types of methods in Java:**

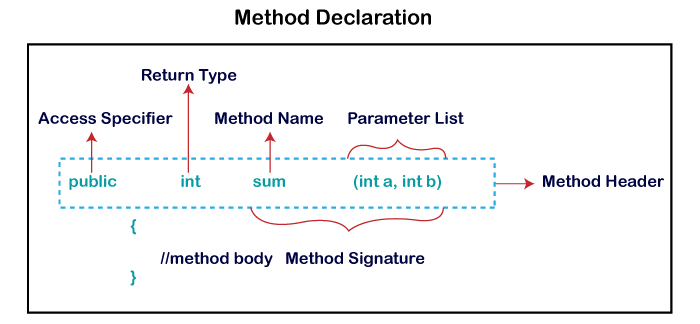
**Standard Library Methods(Print().sqrt())**

**User-defined Methods**

public static void myMethod() {

System.out.println("My Function called");

}



**What are the advantages of using methods?**

1. The main advantage is **code reusability**. We can write a method once, and use it multiple times. We do not have to rewrite the entire code each time. Think of it as, "write once, reuse multiple times".

Traditional code:

**package** week4;

**public** **class** Add {

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

**int** a=5;

**int** b=6;

**int** sum= a+b;

System.***out***.println("Sum of two integer values :"+ sum);

**int** f=10;

**int** h=12;

**int** sum2 = f+h;

System.***out***.println("Sum of two integer values :"+ sum2);

}

}

Using method:

**package** week4;

**class** Addition {

**int** sum = 0;

**public** **int** addTwoInt(**int** a, **int** b){

// adding two integer value.

sum = a + b;

//returning summation of two values.

**return** sum;

}

}

**class** AddMain {

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

// creating an instance of Addition class

Addition add = **new** Addition();

// calling addTwoInt() method to add two integer using instance created

// in above step.

**int** s = add.addTwoInt(1,2);

System.***out***.println("Sum of two integer values :"+ s);

}

}