- **Concept of Class File**

What is a class in Java ?

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

- **Fields**
- **Methods**
- **Constructors**

Syntax to declare a class:

```
class name
{
//data
//methods
}
```

Example:

1. Using Static Method method

```
class Sampleclass{
Static int id;
Static String name;
public static void main(String args[]){
System.out.println(id);
System.out.println(name);
}
}
```

Output: 0
Null

2. Using another class

```
class Student{
int id;
String name;
}
class Sampleclass{
public static void main(String args[]){
Student s1=new Student();
System.out.println(s1.id);
System.out.println(s1.name);
}
}
```

Output: 0
Null

- What is an object in Java ?

An object in Java is a basic unit of Object-Oriented Programming and represents real-life entities.
Objects are instances of a class created to use its attributes and methods.
A typical Java program creates many objects, which as you know, interact by invoking methods.

Once an object is created it takes up space like other variables in memory.

Syntax: **class_name object_name = new class_name();**

Class Name Object Reference DMA Constructor

3 Ways to initialize object

There are 3 ways to initialize object in Java.

1. By reference variable
2. By method
3. By constructor

1) Object and Class Example: Initialization through reference

```
class Student{
int id;
String name;
}
class TestStudent2{
public static void main(String args[]){
  Student s1=new Student();
  s1.id=12;
  s1.name="Darshan";
  System.out.println(s1.id+" "+s1.name);//printing members with a white space  }
}
```

Output: 12

Darshan

- **Difference between Java Class and Objects**

The differences between class and object in Java are as follows:

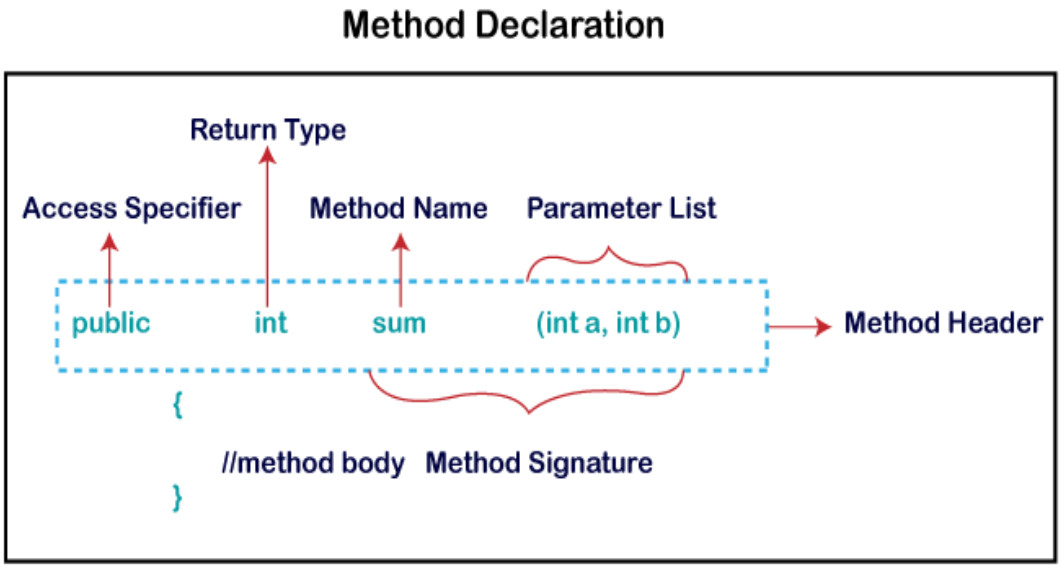
Class	Object
Class is the blueprint of an object. It is used to create objects.	An object is an instance of the class.
No memory is allocated when a class is declared.	Memory is allocated as soon as an object is created.
A class is a group of similar objects.	An object is a real-world entity such as a book, car, etc.
Class is a logical entity.	An object is a physical entity.
A class can only be declared once.	Objects can be created many times as per requirement.
An example of a class can be a car.	Objects of the class car can be BMW, Mercedes, Ferrari, etc.

- ## Method in Java

In general, a method is a way to perform some tasks. Similarly, the method in Java is a collection of instructions that performs a specific task. It provides the reusability of code. We can also easily modify code using methods. In this section, we will learn what is a method in Java, types of methods, method declaration, and how to call a method in Java.

- ## What is a method in Java?

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 easy modification and readability of code, just by adding or removing a chunk of code. The method is executed only when we call or invoke it.



- Access Specifier: Access specifier or modifier is the access type of the method. It specifies the visibility of the method. Java provides **four** types of access specifier:

- **Public:** The method is accessible by all classes when we use public specifier in our application.
- **Private:** When we use a private access specifier, the method is accessible only in the classes in which it is defined.
- **Protected:** When we use a protected access specifier, the method is accessible within the same package or subclasses in a different package.
- **Default:** When we do not use any access specifier in the method declaration, Java uses default access specifier by default. It is visible only from the same package only.

• Naming a Method

While defining a method, remember that the method name must be a verb and start with a lowercase letter. If the method name has more than two words, the first name must be a verb followed by adjective or noun. In the multi-word method name, the first letter of each word must be in uppercase except the first word. For example:

1. Single-word method name: sum(), area()
2. Multi-word method name: areaOfCircle(), stringComparision()

It is also possible that a method has the same name as another method name in the same class, it is known as **method overloading**.