Class

* Defined as a blueprint from which you can create an individual object.
* Class doesn't consume any space.
* Collection of objects
* It is a logical entity

Naming Conventions

It should start with the uppercase letter.

Eg:

public class **Employee**  
{  
//code snippet  
}

Object

* An Object can be defined as an instance of a class.
* An object contains an address and takes up some space in memory.
* Objects can communicate without knowing the details of each other's data or code.

Syntax

List<WebElement> lt = new List<WebElement>

List -> Class

<WebElement> -> Cast ( Data Type)

Lt -> Object

New List -> Creation of new Object for the class.

Naming Conventions

It should start with a lowercase letter

Eg:

BabyDog dog=new BabyDog();

Dog -> Object

Method

A block of code or collection of statements or a set of codes grouped together to perform a certain task or operation.

Advantages

* easy modification
* readability of code
* reusability of code

method declarations have below components in it

* Modifier: It defines the access type of the method i.e., from where it can be accessed in your application. In Java, there 4 types of access specifiers.
* public: It is accessible in all classes in your application.
* protected: It is accessible within the class in which it is defined and in its subclass/es
* private: It is accessible only within the class in which it is defined.
* default: It is declared/defined without using any modifier. It is accessible within the same class and package within which its class is defined.
* The return type: The data type of the value returned by the method or void if does not return a value.
* Parameter list: Comma-separated list of the input parameters is defined, preceded with their data type, within the enclosed parenthesis. If there are no parameters, you must use empty parentheses ().
* Exception list: The exceptions you expect by the method can throw, you can specify these exception(s).
* Method body: it is enclosed between braces. The code you need to be executed to perform your intended operations.

Syntax

public class Main {

public int myMethod(int x, int y)

{

return x + y;

}

}

Int -> Return type

myMethod -> Method name

int x, int y -> Parameters

public -> Access identifier

Naming Conventions

It should start with a lowercase letter.

There are two types of methods in Java:

* **Predefined Method:** In Java, predefined methods are the method that is already defined in the Java class libraries is known as predefined methods. It is also known as the **standard library method** or **built-in method**. We can directly use these methods just by calling them in the program at any point.

**Eg :**

WebDriver driver = New ChromeDriver();

driver.**navigate**()

**Navigate** is the inbuild method for Driver Object

* **User-defined Method:**The method written by the user or programmer is known as **a user-defined** method. These methods are modified according to the requirement.

Eg:

class Adder{

static int **add(**int a,int b){return a+b;}

static int **add(**int a,int b,int c){return a+b+c;}

}

class TestOverloading1{

public static void main(String[] args){

System.out.println(Adder.add(11,11));

System.out.println(Adder.add(11,11,11));

}}

Where **add** is the user defined method