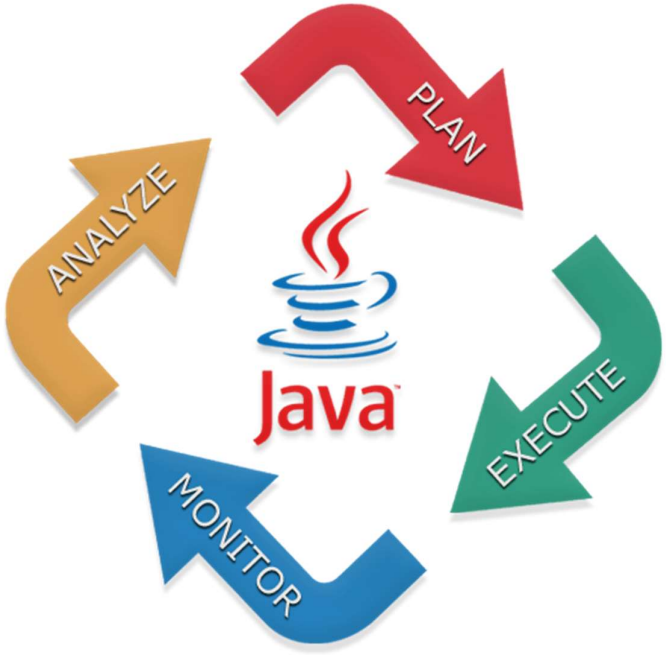
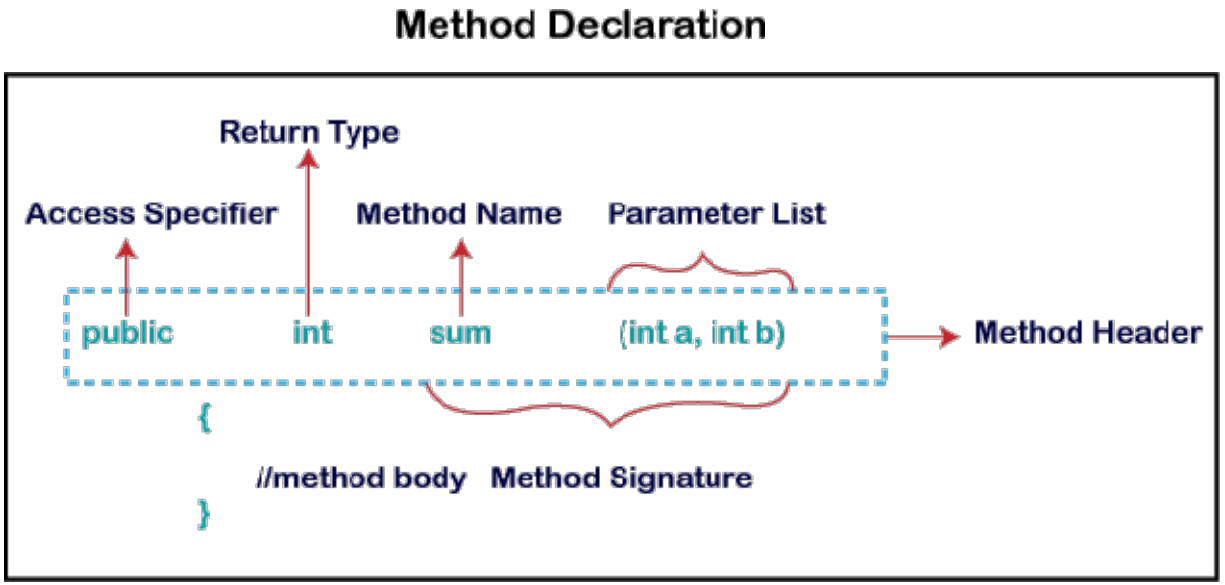


METHODS

The method in Java is a collection of instructions that performs a specific task. It provides the reusability of code.

The method declaration provides information about method attributes, such as visibility, return-type, name, and arguments.



Method Signature: Every method has a method signature. It is a part of the method declaration. It includes the method name and parameter list.

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.

Single-word method name: sum(), area()

Multi-word method name: areaOfCircle(), stringComparision()

There are two types of methods in Java:

- **Predefined Method**
- **User-defined Method**

Predefined Method

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.

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.

Create a User-defined Method to check the number is even or odd.

```
//user defined method
public static void findEvenOdd(int num)
{
//method body
if(num%2==0)
System.out.println(num+" is even");
else
System.out.println(num+" is odd");
}
```

Call or Invoke a User-defined Method

```
import java.util.Scanner;
public class EvenOdd
{
public static void main (String args[])
{
//creating Scanner class object
Scanner scan=new Scanner(System.in);
System.out.print("Enter the number: ");
//reading value from the user
int num=scan.nextInt();
//method calling
findEvenOdd(num);
}
```

Instance Methods

The method of the class is known as an instance method. Before calling or invoking the instance method, it is necessary to create an object of its class.

```
public class InstanceMethodExample
{
public static void main(String [] args)
{
//Creating an object of the class
InstanceMethodExample obj = new InstanceMethodExample();
//invoking instance method
System.out.println("The sum is: "+obj.add(12, 13));
}
int s;
//user-defined method because we have not used static keyword
public int add(int a, int b)
{
s = a+b;
//returning the sum
return s;
}
}
```

There are two types of instance method:

- **Accessor Method**
- **Mutator Method**

Accessor Method: The method(s) that reads the instance variable(s) is known as the accessor method. We can easily identify it because the method is prefixed with the word get. It is also known as getters. It returns the value of the private field. It is used to get the value of the private field.

```
public int getId()
{
return Id;
}
```

Mutator Method: The method(s) read the instance variable(s) and also modify the values. We can easily identify it because the method is prefixed with the word set. It is also known as setters or modifiers. It does not return anything. It accepts a parameter of the same data type that depends on the field. It is used to set the value of the private field.

```
public void setRoll(int roll)
{
this.roll = roll;
}
```

Next Session: Method Overloading