METHODS

METHODS

- The method is a block of instructions that is used to perform a specific task.
- It is used to transfer data.

Syntax to define a method :

[access modifier] [modifier] return type name([datatype var1, datatype var2, …])

{

}

TERMINOLOGIES

1) Method signature

• Method name • Formal argument

2) Method declaration

• Access modifier • Modifier • Return type • Signature

3) Method definition

• Method declaration • Method body / implementation / block

ACCESS MODIFIERS

Access modifiers are used to change the accessibility of a member. We have four levels of access modifiers

1. private
2. default
3. protected
4. public

MODIFIER

They are the keywords which are responsible to modify the characteristics/behaviour of the member. E.g. of modifiers:

1.static
2.abstract
3.final
4.synchronized, etc

RETURN TYPE

- The return type is a data type that specifies what type of data is returned by the method after execution.
- The method after execution can return a value back to the caller.
- Therefore it is mandatory to specify what type of data is returned by the method in the method declaration statement, This is done with the help of return type.

A method can have the following return types

1. void
2. Primitive data type
3. Non-primitive data type

VOID

- void is a data type that is used as a return type when the method returns nothing.
- It is a keyword In java.

NOTE

- A method can’t create inside another method.
- A class can have any number of methods.
- A method will get executed only when it is called, we can call a method with the help of a method call statement.

METHOD CALL STATEMENT

The statement which is used to call a method is known as a method call statement.

Syntax to create a method call statement

methodName([Actual arguments]);

- We can call a no-argument method without passing an actual argument in the method call statement.
- We can call a parameterized-argument method with passing an actual argument in the method call statement.

METHOD CALL STATEMENT FLOW

- Execution of calling method is paused.
- Control is transferred to the called method.
- Execution of called method begins.
- Once the execution of the called method is completed the control is transferred back to the calling method.
- Execution of calling method resumes.

CALLING METHOD

The method which is trying to call another method is known as the calling method (caller).

CALLED METHOD

The method which is being called by the caller is known as a called method.

MAIN METHOD

The execution of a java program always starts from the main method defined as follows:-

```
public static void main(String[] args)
{
 }
```

PURPOSE OF THE MAIN METHOD

- Start the execution
- Control the flow of the execution
- End of execution

NOTE

- A method can be executed only when it is called, we can call a method any number of times, therefore it is said to be code reusability.
- The main method is always called by JVM.

TYPES OF METHODS

Based on number of arguments methods can be classified into 2 types:-

- No argument method
- Parameterized method

NO ARGUMENT METHOD

A method which does not have formal argument is known as no argument method.

Example:-

```
public static void add()
{
    System.out.println(“No argument constructor”);
}
```

PARAMETERIZED METHOD

- The method which has formal argument is known as Parameterized method.
- Parameterized methods are used to accept the data.

FORMAL ARGUMENT

A variable which is declared in the method declaration is known as Formal argument.

ACTUAL ARGUMENT

The value in the method call statement is known as Actual argument.

RETURN STATEMENT

A method after execution will return a data back to the caller with the help of return statement.

return

- return is a keyword.
- It is a control transfer statement.
- When the return statement is executed, the execution of the method is terminated and control is transferred to the calling method.

STEPS TO USE THE RETURN STATEMENT

Step 1: Provide a return type for a method(it should not be void).

Step 2: Use the return statement in the value to be returned.

RULE:- The type specified as return type should be same as the type of value passed in a return statement

FEATURES OF RETURN STATEMENT

1. It is applied at the end of a method, from where it terminates.
2. No statement in the method can be executed after return statement.
3. It can only return a single value from a method to its caller.
4. A function may have more than one termination points, Thus, a number of return statements may be included to terminate the method from a specific point. Eg, if(x>y)

```
        return(x);
    else
        return(y);
```

METHOD OVERLOADING

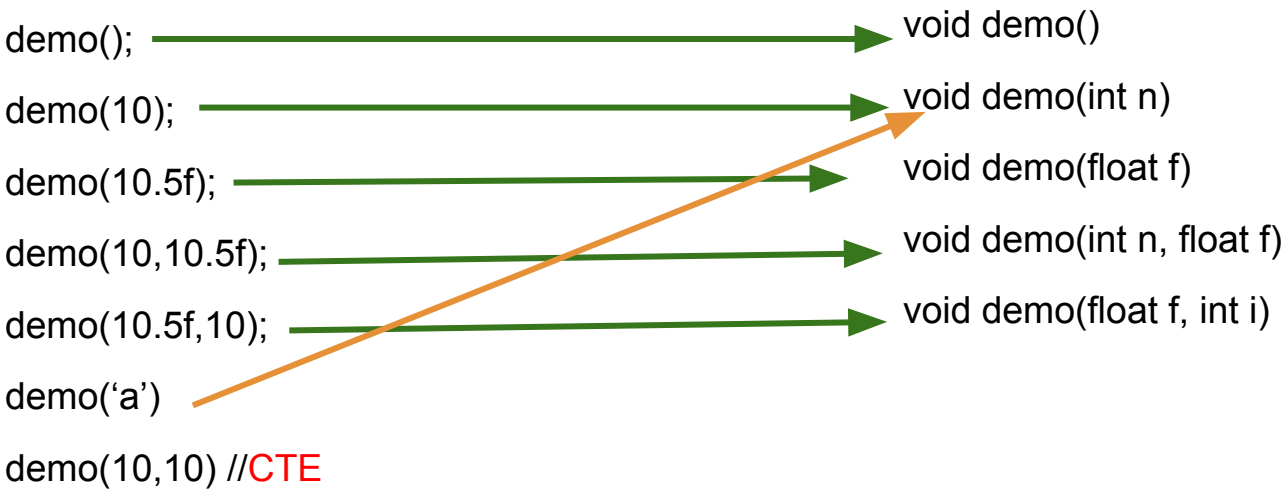
If more than one method is created with the same name but different formal arguments in the same class are known as Method Overloading.

For example:-java.lang. Maths;

abs(int i);
abs(long i);
abs(float i);
abs(double i);

Overloaded methods in Maths class of java.lang package ,having same method names but different formal arguments

EXAMPLE


demo();
demo(10);
demo(10.5f);
demo(10,10.5f);
demo(10.5f,10);
demo('a')
demo(10,10) //CTE
void demo()
void demo(int n)
void demo(float f)
void demo(int n, float f)
void demo(float f, int i)