- In this session we will learn:
  1. Methods
  2. Constructor

Methods

1. A method is a set of statements which are grouped together to perform a specific task.
2. A method is executed when it is called from another method.
3. The main() method is the first method that is executed by JVM (Java Virtual Machine) in Java program.
4. Method Declaration in Java:

   method_modifier return_type method_name(Parameter_list) throws Exceptions

   ```
   {
       // Method body
       // Code to perform the desired task.
       return result;   // return statement (optional)
   }
   ```

Example:

```
// Declaration of instance variables.
    int a = 10;
    int b = 20;
// Declaration of instance method.
   void add()
   {
     int x = a + b;
      System.out.println("Addition of two numbers x = " +x);
    }
```

Points to remember:

1. Variables defines in the method header are called formal parameters or simply parameters.
2. Parameters in a method header are optional; that is, a method may or may not contain parameters.
3. If a method contains parameters, we need to pass values to parameters when a method is invoked. These values are called arguments or actual parameters.
4. A method header represents modifiers, return type value, method name, and parameters of the method.
5. The method name and parameter list together is called method signature.
6. Parameters defined in the method header are always local variables.

Call by Value and Call by Reference in Java

Mechanism to pass values to the methods:

1) **Call By Value** (or pass by value)
2) **Call By Reference**

Call By Value:

1. Call by value (or pass by value) involves passing a copy of the actual value of a variable to a method.
2. This copy can then be modified within the method, but the original value remains unchanged outside the method.

Call by Value and Call by Reference in Java

Mechanism to pass values to the methods:

1) **Call By Value (or pass by value)**
2) **Call By Reference (Pass by Reference)**

Call By Reference :

1. “Call by Reference” means passing a reference (i.e. memory address) of the object by value to a method.
2. A variable of class type contains a reference (i.e. address) to an object, not object itself.
3. Java does not copy the object into the memory. Actually, it copies reference of the object into the memory and passes the copy to the parameters of the called method.

Points to remember:

1. Java passes the arguments both primitives and the copy of object reference by value.

2. Java never passes the object itself.

3. In real-time project, we pass class objects as parameters to method call in Java.

4. Call by value uses copies of data, while call by reference uses memory addresses or references to the actual data.

Constructor in Java

- A constructor is a special member function of a class used to initialize instance variables of a class.
- Constructor is to perform the initialization of data fields of an object in the class with legal values.
- It is automatically invoked when an instance of a class is created using the new operator.

Syntax to Declare Constructor in Java

Syntax:

```
Access modifiers_name class_name(formal_parameter_list)
{
    // Constructor body.
}
```

Example:

```
public class Employee
{
    public Employee()  {
    // constructor code goes here.
  }
}
```

Characteristics of Java Constructor

1. Constructor's name must be exactly the same as the class name in which it is defined.
2. The constructor should not have any return type even void also because if there is a return type then **JVM** would consider as a method, not a constructor.
3. Compiler and **JVM** differentiate constructor and method definitions on the basis of the return type.
4. If You define the method and constructor with the same name as that of the class name then **JVM** would differentiate between them by using return type.
5. Java constructor may or may not contain parameters.
6. A constructor is automatically called and executed by **JVM** at the time of object creation.

Characteristics of Java Constructor

7. **JVM (Java Virtual Machine)** first allocates the memory for variables (objects) and then executes the constructor to initialize instance variables.
8. It is always called and executed only once per object.
9. If we don't define any constructor inside the class, Java compiler automatically creates a default constructor at compile-time and assigns default values for all variables declared in the class.
10. Constructors provide thread safety, meaning no thread can access the object until the execution of constructor is completed.

Different ways to Call Constructor in Java

1. Employee a = new Employee(); // Here, Employee is name of class.
2. new A(); // It is calling A() constructor.
3. super();
4. this();
5. Class.forName(“com.example.com”).newInstance();