Compare and contrast

1. Method and Constructor

Method:-

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 the 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.The most important method in Java is the main() method.

Method Declaration

The method declaration provides information about method attributes, such as visibility, return-type, name, and arguments. It has six components that are known as method header, as shown in the following figure.



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

Method Name: It is a unique name that is used to define the name of a method. It must be corresponding to the functionality of the method. Suppose, if we are creating a method for subtraction of two numbers, the method name must be subtraction(). A method is invoked by its name.

Method Body: It is a part of the method declaration. It contains all the actions to be performed. It is enclosed within the pair of curly braces.

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.

There are two types of methods in Java:

1.Predefined Method

2.User-defined Method

1.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. Some pre-defined methods are length(), equals(), compareTo(), sqrt(), etc. When we call any of the predefined methods in our program, a series of codes related to the corresponding method runs in the background that is already stored in the library.Each and every predefined method is defined inside a class. Such as print() method is defined in the java.io.PrintStream class. It prints the statement that we write inside the method. For example, print(“Java”), it prints Java on the console.

2.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.

Constructor:-

Java constructors or constructors in Java is a terminology been used to construct something in our programs. A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes.

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling the constructor, memory for the object is allocated in the memory. It is a special type of method which is used to initialize the object. Every time an object is created using the new() keyword, at least one constructor is called.

The rules for writing constructors are as follows:

\*Constructor (s) of a class must have the same name as the class name in which it resides.

\*A constructor in Java can not be abstract, final, static, or Synchronized. \*Access modifiers can be used in constructor declaration to control its access i.e which other class can call the constructor.

Types of Constructors in Java

1.No-argument constructor

2. Paramrterized

1. No-argument constructor: A constructor that has no parameter is known as the default constructor. If we don’t define a constructor in a class, then the compiler creates a default constructor(with no arguments) for the class. And if we write a constructor with arguments or no arguments then the compiler does not create a default constructor.
2. Parameterized Constructor: A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with our own values, then use a parameterized constructor.
3. Constructor and destructor

Constructor:-

Java constructors or constructors in Java is a terminology been used to construct something in our programs. A constructor in Java is a special method that is used to initialize objects. The constructor is called when an object of a class is created. It can be used to set initial values for object attributes.

In Java, a constructor is a block of codes similar to the method. It is called when an instance of the class is created. At the time of calling the constructor, memory for the object is allocated in the memory. It is a special type of method which is used to initialize the object. Every time an object is created using the new() keyword, at least one constructor is called.

The rules for writing constructors are as follows:

\*Constructor (s) of a class must have the same name as the class name in which it resides.

\*A constructor in Java can not be abstract, final, static, or Synchronized. \*Access modifiers can be used in constructor declaration to control its access i.e which other class can call the constructor.

Types of Constructors in Java

1. No-argument constructor: A constructor that has no parameter is known as the default constructor. If we don’t define a constructor in a class, then the compiler creates a default constructor(with no arguments) for the class. And if we write a constructor with arguments or no arguments then the compiler does not create a default constructor.

Example:

1. Parameterized Constructor: A constructor that has parameters is known as parameterized constructor. If we want to initialize fields of the class with our own values, then use a parameterized constructor.

Destructor

When we create an object of the class it occupies some space in the memory (heap). If we do not delete these objects, it remains in the memory and occupies unnecessary space that is not upright from the aspect of programming. To resolve this problem, we use the destructor. In this section, we will discuss the alternate option to the destructor in Java. Also, we will also learn how to use the finalize() method as a destructor.

The destructor is the opposite of the constructor. The constructor is used to initialize objects while the destructor is used to delete or destroy the object that releases the resource occupied by the object.Remember that there is no concept of destructor in Java. In place of the destructor, Java provides the garbage collector that works the same as the destructor. The garbage collector is a program (thread) that runs on the JVM. It automatically deletes the unused objects (objects that are no longer used) and free-up the memory. The programmer has no need to manage memory, manually. It can be error-prone, vulnerable, and may lead to a memory leak.

Destructor working

When the object is created it occupies the space in the heap. These objects are used by the threads. If the objects are no longer is used by the thread it becomes eligible for the garbage collection. The memory occupied by that object is now available for new objects that are being created. It is noted that when the garbage collector destroys the object, the JRE calls the finalize() method to close the connections such as database and network connection.

From the above, we can conclude that using the destructor and garbage collector is the level of developer’s interference to memory management. It is the main difference between the two. The destructor notifies exactly when the object will be destroyed. While in Java the garbage collector does the same work automatically. These two approaches to memory management have positive and negative effects. But the main issue is that sometimes the developer needs immediate access to memory management.