engineering.vips.edu

VIPS Technical Campus
योगः कर्मसु कौशलम्
IN PURSUIT OF PERFECTION

SCHOOL OF ENGINEERING AND TECHNOLOGY

Course : OBJECT ORIENTED PROGRAMMING

Paper Code: AIDS 202,AIML202,IIOT202

Faculty : Dr. Shivanka
Assistant Professor
VIPS

Constructors

- A constructor initializes an object immediately upon creation.
- It has the same name as the class in which it resides and is syntactically similar to a method.
- Once defined, the constructor is automatically called immediately after the object is created, before the new operator completes. Constructors look a little strange because they have no return type, not even void. This is because the implicit return type of a class' constructor is the class type itself.
- It is the constructor's job to initialize the internal state of an object so that the code creating an instance will have a fully initialized, usable object immediately.

Rules for creating Java constructor

- There are two rules defined for the constructor.

- Constructor name must be the same as its class name
- A Constructor must have no explicit return type
- A Java constructor cannot be abstract, static, final, and synchronized.
- 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.

Types of Java constructors

- There are three types of constructors in Java:

- Default constructor (no-arg constructor):A constructor is called "Default Constructor" when it doesn't have any parameter.
- Parameterized constructor:The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.
- Copy Constructor:There is no copy constructor in Java. However, we can copy the values from one object to another like copy constructor in C++.

Default constructor (no-arg constructor)

Java Default Constructor

A constructor is called "Default Constructor" when it doesn't have any parameter.

Syntax of default constructor:

```
<class_name>(){}
```

Example of default constructor

- In this example, we are creating the no-arg constructor in the Student class. It will be invoked at the time of object creation.

```
//Java Program to create and call a default constructor
class Student1{
//creating a default constructor
    Student1(){System.out.println("class is created");}
//main method
public static void main(String args[]){
//calling a default constructor
Student1 b=new Student1();
} }
```

Output:

```
class is created
```

- Note : If there is no constructor in a class, compiler automatically creates a default constructor.

Java default constructor

- Q) What is the purpose of a default constructor?

The default constructor is used to provide the default values to the object like 0, null, etc., depending on the type.

Example of default constructor that displays the default values

```
////Let us see another example of default constructor which displays the default values
class Student2{
int id;
String name;
//method to display the value of id and name
void display(){System.out.println(id+" "+name);}

public static void main(String args[]){
//creating objects
Student2 s1=new Student2();
Student2 s2=new Student2();
//displaying values of the object
s1.display();
s2.display();
} }
```

Output:

- 0 null
- 0 null
- Explanation:In the above class,you are not creating any constructor so compiler provides you a default constructor. Here 0 and null values are provided by default constructor.

Java Parameterized Constructor

- A constructor which has a specific number of parameters is called a parameterized constructor.
- Why use the parameterized constructor?
- The parameterized constructor is used to provide different values to distinct objects. However, you can provide the same values also.
- 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.

Example of Parameterized constructor.

```
import java.io.*;
class Student {
    // data members of the class.
    String name;
    int id;
    Student(String name, int id)
    {
        this.name = name;
        this.id = id;
    }
}
```

```
class stud {
    public static void main(String[] args)
    {
        // This would invoke the parameterized
constructor.
        Student s = new Student("Gourav", 68);
        System.out.println("StudentName :" +
s.name  + " and StudentId : " + s.id);
    }
}
```

Java Program to demonstrate the use of the parameterized constructor.

SCHOOL OF ENGINEERING AND TECHNOLOGY

```
class Student3{
    int id;
    String name;
    //creating a parameterized constructor
    Student3(int i,String n){
    id = i;    name = n;   }
    //method to display the values
    void display(){System.out.println(id+" "+name);}
   public static void main(String args[]){
    //creating objects and passing values
    Student3 s1 = new Student3(101,"Pooja");
    Student3 s2 = new Student3(201,"Shivam");
    //calling method to display the values of object
s1.display();
    s2.display();
   }  }
```

Output:

```
101 Pooja
201 Shivam
```

Java Copy Constructor

There is no copy constructor in Java. However, we can copy the values from one object to another like copy constructor in C++.

- There are many ways to copy the values of one object into another in Java. They are:

- By constructor
- By assigning the values of one object into another
- By clone() method of Object class

- Unlike other constructors copy constructor is passed with another object which copies the data available from the passed object to the newly created object.

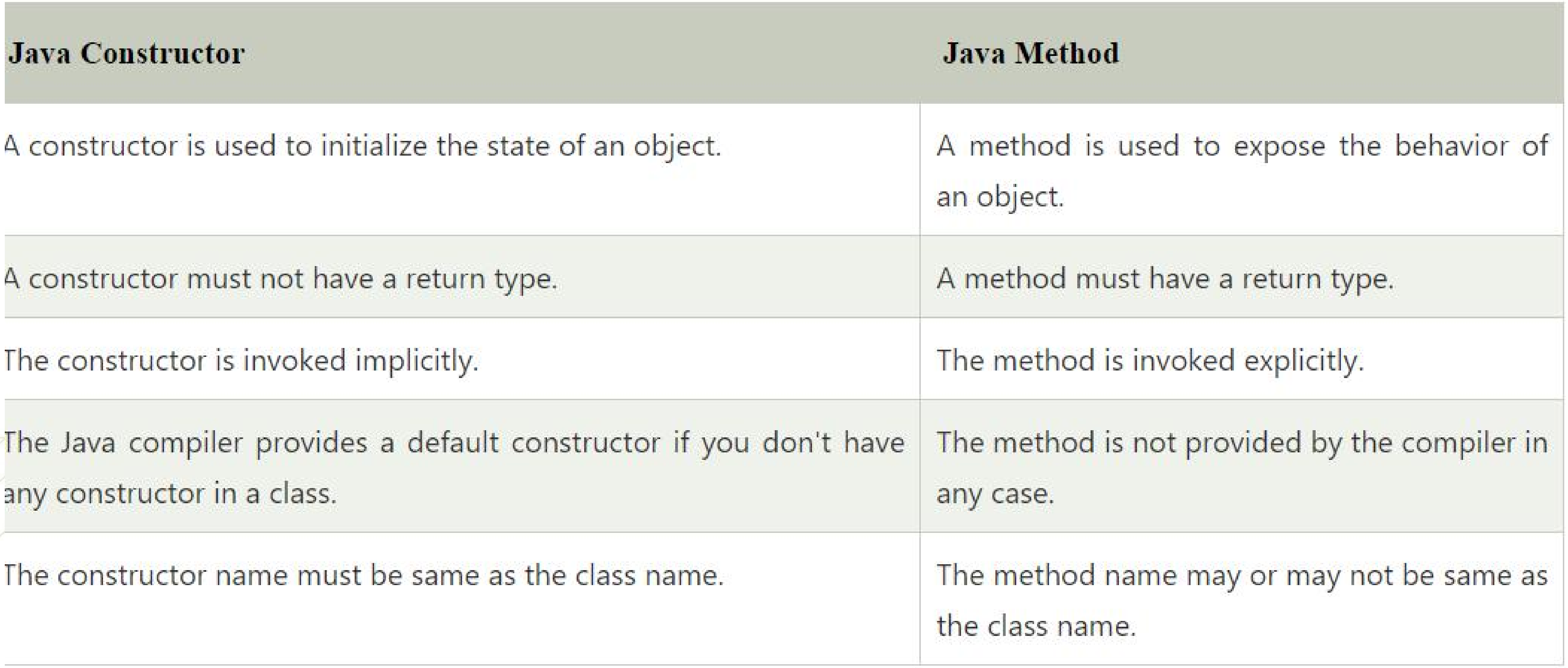
- Note: In Java,there is no such inbuilt copy constructor available like in other programming languages such as C++, instead we can create our own copy constructor by passing the object of the same class to the other instance(object) of the class.

//Java program to initialize the values from one object to another object.

```
   class Student6{
int id;  String name;
   //constructor to initialize integer and string
   Student6(int i,String n){
   id = i;
   name = n;   }
   //constructor to initialize another object
   Student6(Student6 s){
   id = s.id;
   name =s.name;   }
```

```
  void display(){System.out.println(id+" "+name);}
   public static void main(String args[]){
    Student6 s1 = new Student6(111,"Karan");
    Student6 s2 = new Student6(s1);
    s1.display();
    s2.display();
   }
}
```

Difference between Java Constructor and Java Method



Java Constructor	Java Method
A constructor is used to initialize the state of an object.	A method is used to expose the behavior of an object.
A constructor must not have a return type.	A method must have a return type.
The constructor is invoked implicitly.	The method is invoked explicitly.
The Java compiler provides a default constructor if you don't have any constructor in a class.	The method is not provided by the compiler in any case.
The constructor name must be same as the class name.	The method name may or may not be same as the class name.

Thank You