*Constructor*

* 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.
* every time an object is created using the new () keyword, at least one constructor is called.
* It does not return anything even void also.
* constructor cannot be abstract, static, final.
* When you write any constructor in the class then default constructor will not added by JVM.

There are two types of constructor are as

1. *Default constructor* (No- argument constructor)

A constructor that does not accept any arguments called as default constructor.

Example

**package** com.velocity;

**public** **class** Student {

String city;

**public** Student() {

city = "pune";

}

**public** **static** **void** main(String[] args) {

Student student = **new** Student();

System.***out***.println("Name of city is >> " + student.city);

}

}

1. Parameterized constructor

A constructor with arguments called as parameterized constructor.

Example

**package** com.velocity;

**public** **class** Student {

**int** rollNo;

String name;

String city;

**public** Student(**int** studentRollNo, String studentName, String studentCity) {

rollNo = studentRollNo;

name = studentName;

city = studentCity;

System.***out***.println("roll number of student >> " + rollNo);

System.***out***.println("Name of student >> " + name);

System.***out***.println("city of of student >> " + city);

}

**public** **static** **void** main(String[] args) {

Student student = **new** Student(1, "Rahul", "Pune");

}

}

Output

roll number of student >> 1

Name of student >> Rahul

city of of student >> Pune

*point to be remember*

* main purpose constructor is to initialization of global variables at the time of at object creation.
* Constructor cannot be overridden because we can’t write multiple constructors with same arguments. If you are tried to write it then you will get compiler time error.
* When you don’t write any constructor in the class then default constructor will be added by JVM automatically at the compile time.
* When you write any constructor in the class then default constructor will not added by JVM.
* Constructor does not have any return type. If you write any return type then it will be called as method.

we can call the constructor following ways

Student student = **new** Student () ;// by object creation

super();//upcoming lecture

this();//upcoming lecture

new Employee();//upcoming lecture

Constructor can be overloaded because we can write same name with different arguments.

class.forName(“com.test”).newInstance();

**package** com.velocity;

**public** **class** Student {

**int** rollNo;

String name;

String city;

**public** Student(**int** studentRollNo, String studentName, String studentCity) {

rollNo = studentRollNo;

name = studentName;

city = studentCity;

}

**public** Student(**int** studentRollNo, String studentName) {

rollNo = studentRollNo;

name = studentName;

}

@Override

**public** String toString() {

**return** "Student [rollNo=" + rollNo + ", name=" + name + ", city=" + city + "]";

}

**public** **static** **void** main(String[] args) {

Student student = **new** Student(1, "Rahul", "Pune");

Student student1 = **new** Student(1, "Rahul");

System.***out***.println(student);

System.***out***.println(student1);

}

}

Output

1, Rahul, Pune

1, Rahul, null