Constructor

Daste: 29th Aug 2024

A constructor in Java is a special member of class that is used to initialize global objects/variables.
The constructor is called when an object of a class is created.

At the time of constructor declaration below points need to follow:

1. Constructor name should be same as class name.
2. you should not declare any return type for the constructor (like void).
3. Any no of constructor can be declared in a java class but constructor name should be same as class name, but arguments/parameter should be different--> Constructor overloading.

Use of Constructor

1. To copy/load all members of class into object --> when we create object of class
2. To initialize data member/variable

Types of Constructors

1. Default Constructor
2. User defined Constructor

1. Default Constructor

If Constructor is not declared in java class then at the time of compilation compiler will provide Constructor for the class
If programmer has declared the constructor in the class then compiler will not provided default Constructor.
The Constructor provided by compiler at the time of compilation is known as Default Constructor

2. User defined Constructor

If programmer is declaring constructor in java class then it is considered to be as User defined constructor.
User defined Constructor are classified into 2 types

1. Without/zero parameter constructor
2. With parameter constructor

1: Example of Default Constructor

```
package Constructor;
public class Sample1
{
	//1: example of Default Constructor

	//default constructor -> provided by compiler
	//use: to copy all the members of class into object
//	Sample1()
//	{
//
//	}

	public void m1()
	{
		System.out.println("method m1 from same class");
	}

	public static void main(String[] args)
	{
		Sample1 s1=new Sample1();    //default constructor call from same class
		s1.m1();

		//1: Sample1 --> className --> as a dataType
		//2: s1 -> objectName -> to identify/refer object
		//3: new -> keyword  -> to create blank/empty object
		//4: Sample1() -> className() -> Constructor call

		Sample2 s2=new Sample2();    //default constructor call from diff class
		s2.m2();
	}
}
```

```
package Constructor;
public class Sample2
{
	//default constructor  -> provided by compiler
	//use: to copy all the members of class into object
//	Sample2()
//	{
//
//	}



	public void m2()
	{
		System.out.println("method m2 from diff class");
	}
}
```

2: User defined constructor

2.1 user defined without/zero parameter

```
package Constructor;
public class Sample3
{
	//2: User defined constructor without parameter

	//1: variable declaration
	int num1;   //10
	int num2;   //20


	//2: initialize variable
	//user defined constructor -> provided by
	//use1: to initialize global variable
	//use2: to copy all the members of class into object
	Sample3()      //without/zero parameter constructor
	{
		num1=10;
		num2=20;
	}


	//3: variable usage
	public void add()
	{
		System.out.println(num1+num2);
	}

	public void mult()
	{
		System.out.println(num1*num2);
	}


	public static void main(String[] args)
	{
		Sample3 s3=new Sample3();     //user defined constructor call from same class
		s3.add();
		s3.mult();

		System.out.println("--------");

		Sample4 s4=new Sample4();
		s4.sub();
	}
}
```

```
package Constructor;
public class Sample4
{
	int num3;  //50
	int num4;  //30

	//user defined constructor -> provided by user
	//use1: to initialize global variable
	//use2: to copy all the members of class into object
	Sample4()	//without/zero parameter constructor
	{
		num3=50;
		num4=30;
	}

	public void sub()
	{
		System.out.println(num3-num4);
	}
}
```

2.2 user defined with parameter

```
package Constructor;
public class Sample5
{
	//3: user defined with parameter constructor

	int num1;	//100
	int num2;	//200

	//constructor with 2 int(int int) parameter
	Sample5(int a, int b)	//a=100, b=200
	{
		num1=a;  //100	//globalVariable=localVariable -> assign local variable info into global variable
		num2=b;  //200
	}

	public void add()
	{
		System.out.println(num1+num2);
	}

	public void mult()
	{
		System.out.println(num1*num2);
	}


	public static void main(String[] args)
	{
		Sample5 s5=new Sample5(10, 20);
		s5.add();
		s5.mult();

		System.out.println("------");

		Sample5 s6=new Sample5(50, 60);
		s6.add();
		s6.mult();

		System.out.println("----------------");

		Sample6 s7=new Sample6(5, 9);
		s7.sub();

		Sample6 s8=new Sample6(100, 80);
		s8.sub();
	}
}
```

```
package Constructor;
public class Sample6
{
		int num3;     //5
		int num4;    //9

		//with 2 int parameter cosntructor
		Sample6(int c, int d)   //c=5, d=9
		{
			num3=c;   //5
			num4=d;   //9
		}

		public void sub()
		{
			System.out.println(num3-num4);   //5-9 = -4
		}
}
```

Example of constructor overloading

```
package Constructor;
public class Sample7
{
	//4: constructor overloading

	int num1;   //50
	int num2;   //60
	String sname;    //Uma

	Sample7()   //constructor without parameter
	{
		num1=10;
		num2=20;
	}

	Sample7(int a, int b)   //a=50 , b=60 //constructor with 2 int parameter
	{
		num1=a;    //50
		num2=b;    //60
	}


	Sample7(String s1)  // s1=Uma  //constructor with String parameter
	{
		sname=s1;   //Uma
	}

	public void add()
	{
		System.out.println(num1+num2);
	}

	public void mult()
	{
		System.out.println(num1*num2);
	}

	public void studentName()
	{
		System.out.println(sname);
	}

	public static void main(String[] args)
	{
		Sample7 s7=new Sample7();
		s7.add();
		s7.mult();

		System.out.println("---");

		Sample7 s8=new Sample7(50, 60);
		s8.add();
		s8.mult();

		System.out.println("--------");

		Sample7 s9=new Sample7("Uma");
		s9.studentName();
	}
}
```