Constructor:

Date: 13th March 24

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 provide 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

Example1: default constructor

```
public class Sample1
{
	//Example1: default constructor

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


	public void m1()
	{
		System.out.println("running method m1");
	}


	public void m2()
	{
		System.out.println("running method m2");
	}


	public static void main(String[] args)
	{
		//1: default constructor call from same class
		Sample1 s1=new Sample1();
		s1.m1();
		s1.m2();

		//1: sample1-> className  -> datatype
		//2: s1    -> objectName -> to refer/identify object
		//3: new   -> keyword    -> use to create blank/empty object
		//4: Sample1() -> className() -> constructor call  -> to copy/load all
the members of class into object

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

		//2: default constructor call from diff class
		Sample2 s2=new Sample2();
		s2.m3();
	}
}



public class Sample2
{

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


	public void m3()
	{
		System.out.println("running method m3 from diff class");
	}
}
```

Date: 14th March 23

2. User defined Constructor

If programmer is declaring constructor in java class, then it is considered to be as User defined constructor.

Example2: User defined Without/Zero parameter constructor

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

	//step1: variable declaration
	int num1;	//num1=10
	int num2;	 //num2=20


	//step2: variable initialization
	//user defined constructor --> provided by user
	//use1: to initialize global variable
	//use2: copy/load all the members of class into object
	Sample3()	//without/zero parameter user defined constructor
	{
		num1=10;	//globalVariable= info
		num2=20;
	}


	//step3: variable usage
	public void add()
	{
		System.out.println(num1+num2);
	}

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


	public static void main(String[] args)
	{
		//3: user defined constructor call from same class
		Sample3 s3=new Sample3();
		s3.add();	//30
		s3.mul();	//200

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

		//4: user defined constructor call from same class
		Sample4 s4=new Sample4();
		s4.sub();	//  10
	}
}
```

```java
package Constructor;
public class Sample4
{
	//step1: variable declaration
	int num3;  //70
	int num4;  //60


	//step2: variable initialization
	//user defined constructor
	//use1:  to initialize global variable
	//use2: copy/load all the members of class into object
	Sample4()        //without/zero parameter constructor
	{
		num3=70;
		num4=60;
	}


	//step3: variable usage
	public void sub()
	{
		System.out.println(num3-num4);
	}
}
```

Example3: Constructor with parameter

Date: 15th March 2024

```
package Constructor;
public class Sample5
{
	//Example3: constructor with parameter

	int num1;	//50
	int num2;	//60

	//user defined constructor
	//use1: initialize global variable
	//use2: copy all the members of class into object
	Sample5(int a, int b)  //50, 60 //constructor with int, int parameter
	{
		num1=a;  //50  //globalVariable=localVariable
				//assign local variable info into global variable
		num2=b;		//60
	}


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

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


	public static void main(String[] args)
	{
		Sample5 s5=new Sample5(10,20);
		s5.add();	//30
		s5.mul();	//200

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

		Sample5 s6=new Sample5(50, 60);
		s6.add();  //110
		s6.mul();  //3000

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

		Sample6 s7=new Sample6(800, 700);
		s7.sub();	//100

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

		Sample6 s8=new Sample6(1600, 500);
		s8.sub();	//1100
	}
}
```

```java
package Constructor;

public class Sample6
{
	int num3;	//800
	int num4;	//700

	//user defined with parameter constructor
	Sample6(int a, int b)	//800,  700
	{
		num3=a;	//800
		num4=b;		//700
	}

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

Example4: Constructor Overloading

Date: 18th Mar 24

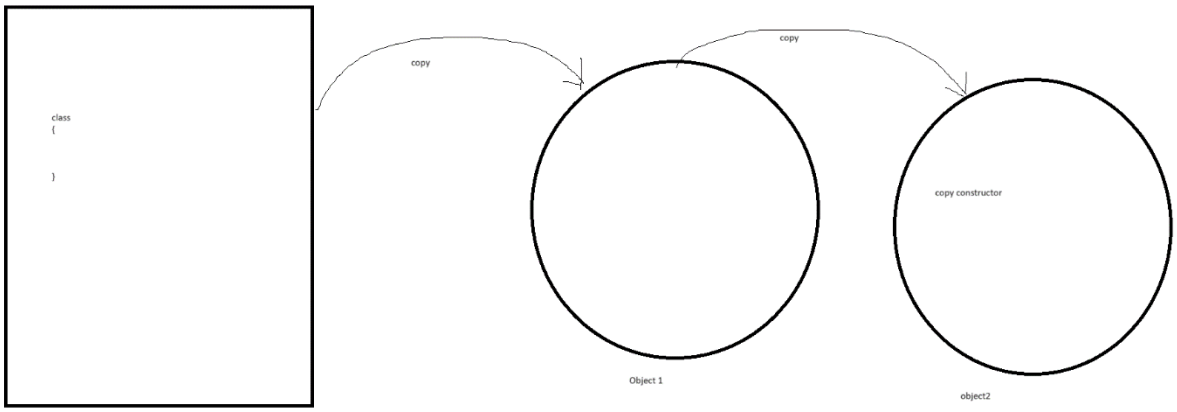
Declaring multiple constructors with same constructor name, but with different parameter in a same class is called constructor overloading

```
package Constructor;
public class Sample7
{
	//example4: Multiple constructor in same class : Constructor
Overloading

	int num1;   //50
	int num2;    //60
	String name;   //Amol

	//user defined -> without/zero parameter constructor
	Sample7()
	{
		num1=10;
		num2=20;
	}

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

	//user defined with String parameter
	Sample7(String s1)    //Amol
	{
		name = s1;    //Amol
	}


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

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

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

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

```
            Sample7 s7=new Sample7();
            s7.add();    //30
            s7.mul();    //200

            System.out.println("----");
            Sample7 s8=new Sample7(50, 60);
            s8.add();
            s8.mul();

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

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

Example5: Copy Constructor
Date: 19th March 2024

```
package Constructor;
public class Sample8
{
	//5: example of copy constructor

	int num1;
	int num2;

	Sample8(int a, int b)
	{
		num1=a;
		num2=b;
	}

	//copy constructor
	Sample8(Sample8 s5)
	{
		num1=s5.num1;
		num2=s5.num2;
	}


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


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


	public static void main(String[] args)
	{

		Sample8 s1=new Sample8(10, 20);
		s1.add();   //30
		s1.mult();   //200

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

		Sample8 s2=new Sample8(s1);
		s2.add();   //30
		s2.mult();  //200

	}
}
```



Interview Questions:

1: What is constructor? types of constructor?
2: What is default constructor?
3: What is user defined constructor?
4: What is a parameterized constructor?
A constructor that contains one or more parameters.
5: When a constructor is called/invoked in Java?
6: Is it possible for a class to have multiple constructors in java?
7: Can we Overload Constructor?
8: What is Constructor Overloading?
9: Can we have more than one constructor with same signature in a class?
No, we cannot have more than one constructor with same signature in a class.
10. Does a constructor return any value? No
11: What happens if you keep a return type for a constructor?
Constructor must not have a return type.
if we write return type with constructor, It will be treated as a normal method.
12: What is the difference between constructor and method?
13: what is copy constructor?
A constructor which is used to copy the data of one object to another object.
14: Do we have destructors in Java?
No. Destructors are usually used to deallocate memory and do other cleanup for a class object
15. Give an example that proves the definition of constructor.
public class Test

```
    {
          String name;
         int age;

         // constructor function.
         Test()
         {
            name = "John";
            age = 20;
         }
    }
```

Logical Questions on Constructor

1: Will the below code compile successfully? If not, why?

```
class Test
 {
   Test()
   {

    }

  public void display()
  {
      Test();
  }
}
```

---> No

2: Will the below code compile successfully?

```
class Test
 {
   Test()
   {

   }

  public void display()
  {
      new Test();
  }
}
```

---> Yes