Methods:

Date: 17th Aug 2024

A method is a block of code which only runs when it is called.
Methods are used to perform certain actions, and they are also known as functions.

Why use methods? To reuse code: define the code once, and use it many times.

1. main method (pre-defined)

In any Java program, the main() method is the starting point from where compiler starts program execution. So, the compiler needs to call the main() method.

without main method we can't run any java program.

2. Regular method (user defined)

1. static regular method

1. static regular method call from same class --> methodname();
2. static method call from different/another class -->diffClassName.methodname();

2. non- static regular method

3. non-static method call from same class --> 1. create object of same class 2. objectname.methodname();
4. non-static method call from different/another class --> 1. create object of diff class 2. diffClassObjectName.methodname();

1: static method call from same class

```
package Methods;
public class Sample1
{
	//1: static method call from same class

	//main method
	public static void main(String[] args)   //pre defined method
	{
		System.out.println("Hi");

		System.out.println("hello");

		m1();			//call method  -> methodname();
		m2();
		m2();	//method Reuse
	}


	//static ->regular method
	public static void m1()	//user defined
	{
		System.out.println("running static regular method m1 from same class");
	}


	//static ->regular method
	public static void m2()	//user defined
	{
		System.out.println("running static regular method m2 from same class");
	}
}
```

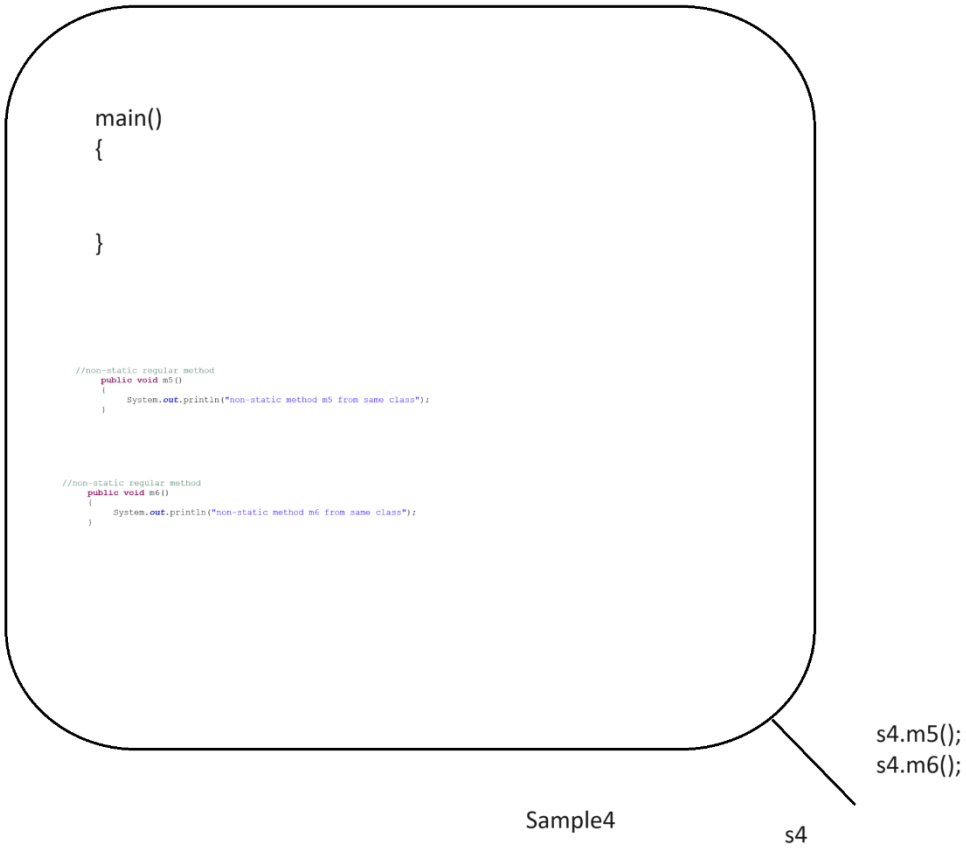
2: static method call from diff class

```
package Methods;
public class Sample2
{
	//2: static regular method class from diff class

	public static void main(String[] args)
	{
		Sample3.m3();			//diffclass.methodName();
		Sample3.m4();
		Sample3.m4();	//method reuse
	}
}
```

```
package Methods;
public class Sample3
{
	public static void m3()
	{
		System.out.println("running static method m3 from diff class");
	}

	public static void m4()
	{
		System.out.println("running static method m4 from diff class");
	}
}
```

3: non-static method call from same class

```
package Methods;
public class Sample4
{
	//3: non-static method call from same class
	public static void main(String[] args)
	{
		//1: create object of same/current class
		//2: method call -> objectName.methodName();

		//className objectName=new className();	-> syntax of object creation

		Sample4 s4=new Sample4();
		s4.m5();
		s4.m6();
		s4.m6();

		//1: Sample4 -> className	-> as  a dataType (objectType)
		//2: s4	-> objectName	-> use to identify/refer object
		//3: new	-> keyword	-> use to create blank/empty object
		//4: Sample4()	-> className() -> constructor -> use to copy all the members of class into object
	}

	//non-static regular method
	public void m5()
	{
		System.out.println("non-static method m5 from same class");
	}

	//non-static regular method
	public void m6()
	{
		System.out.println("non-static method m6 from same class");
	}
}
```



4: non-static method call from diff class

```
package Methods;
public class Sample5
{
	//4: non-static method call from diff class

	public static void main(String[] args)
	{
		Sample6 s6=new Sample6();   //1: create object of diff class
		s6.m7();		//2: method call -> diffClassObjectName.methodname();
		s6.m8();
		s6.m8();
	}
}


package Methods;
public class Sample6
{
	public void m7()
	{
		System.out.println("running non-static regular method m7 from diff class");
	}


	public void m8()
	{
		System.out.println("running non-static regular method m8 from diff class");
	}
}
```

5: method without/zero parameter

```
package Methods;
public class Sample7
{
	public static void main(String[] args)
	{
		//1: static method call from same class
		m1();

		//2: static method call from diff class
		Sample8.m2();

		//3: non-static method call from same class
		Sample7 s7=new Sample7();
		s7.m3();


		//4: non-static method call from diff class
		Sample8 s8=new Sample8();
		s8.m4();
	}

	public static void m1()  //without/zero parameter method
	{
		System.out.println("static method m1 from same class");
	}

	public void m3()   //without/zero parameter method
	{
		System.out.println("non-static method m3 from same class");
	}
}


package Methods;
public class Sample8
{
	public static void m2()   //without/zero parameter method
	{
		System.out.println("static method m2 from diff class");
	}


	public void m4()   //without/zero parameter method
	{
		System.out.println("non-static method m4 from diff class");
	}
}
```

6: method with parameter

```
package Methods;
public class Sample9
{
	//6: method with parameter
	public static void main(String[] args)
	{
		add(10,20);
		add(5,6);
		add(70,80);

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

		Sample9 s9=new Sample9();
		s9.squareOfNum(4);
		s9.squareOfNum(9);
	}

	//method 2 int (int, int) parameter
	public static void add(int num1, int num2) // num1=70,  num2=80
	{
		System.out.println(num1+num2);   //70+80
	}

	public void squareOfNum(int a)   //a=9;
	{
		System.out.println(a*a);    //9*9=
	}
}
```

```
package Methods;
public class Sample10
{
	public static void main(String[] args)
	{
		studentName("rahul");
		studentName("amol");

		studentFullName("abc1", "xyz1");
		studentFullName("abc12", "xyz2");

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

		Sample11.studentInfo("Amol", 101, 67.5f, 'A');
		Sample11.studentInfo("Tushar", 102, 58.2f, 'B');
	}

	//method with String parameter
	public static void studentName(String name)   //name= rahul
	{
		System.out.println(name);
	}

	//method with String,String(2 String) parameter
	public static void studentFullName(String fn, String ln)
	{
		System.out.println(fn+", "+ln);
	}
}
```

```
package Methods;
public class Sample11
{
	//method with String,int,float, char parameter
	public static void studentInfo(String sName,int sRollNum, float sPer, char sGrade)
	{
		System.out.println("Name: "+sName);
		System.out.println("RollNum: "+sRollNum);
		System.out.println("Percentage: "+sPer+"%");
		System.out.println("Grade: "+sGrade);
	}
}
```