DAY-31

27 July 2023 15:35

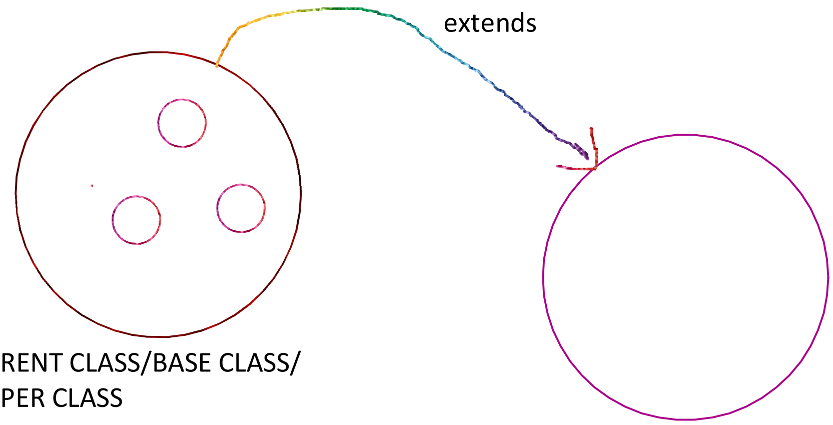
INHERITANCE

WHAT IS INHERITANCE ?

- IT IS A PROCESS OF DERIVING THE PROPERTIES OF ONE CLASS FROM ANOTHER CLASS
- THE CLASS FROM WHICH PROPERTIES ARE INHERITED IS KNOWN AS PARENT CLASS/BASE CLASS/ SUPER CLASS
- THE CLASS TO WHICH PROPERTIES ARE INHERITED IS CALLED CHILD CLASS / DERIVED CLASS/ SUB CLASS
- WE CAN ESTABLISH THE RELATIONSHIP BETWEEN CLASS BY USING **extends**
- INHERITANCE IS ALSO CALLED **IS-A RELATIONSHIP**

NOTE :

EXTENDS : IT IS KEYWORD WHICH IS USED TO DERIVE THE PROPERTIES FROM THE PARENT CLASS TO THE CHILD CLASS. THIS IS USED IN INHERITANCE



PARENT CLASS/BASE CLASS/
SUPER CLASS

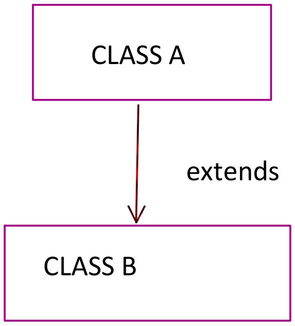
DERIVED CLASS/SUB CLASS/ CHILD CLASS

TYPES OF INHERITANCE

- SINGLE LEVEL INHERITANCE
- MULTI LEVEL INHERITANCE
- HIERARICHAL INHERITANCE
- MULTIPLE INHERITANCE
- HYBRID INHERITANCE

SINGLE LEVEL INHERITANCE

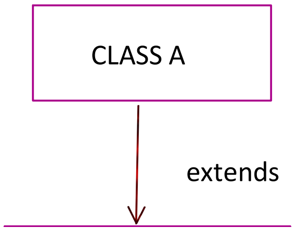
- IN THIS TYPE OF INHERITANCE , WE WILL HAVE A SINGLE SUPER CLASS AND SINGLE SUB CLASS

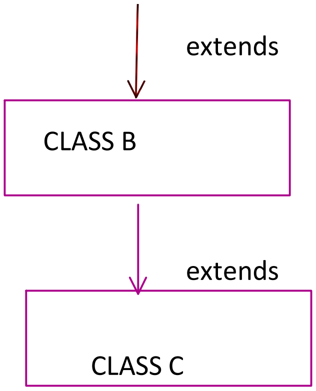


```
class Test1
{
        static int a = 10;
        float b = 23.5f;

        public void m1()
        {
                System.out.println("in parent class method");
        }

}

class Test2 extends Test1
{
        public static void main(String[] args)
        {
                        Test2 a1 = new Test2();
                        System.out.println(a);
                        System.out.println(a1.b);
                        a1.m1();

        }
}
```

MULTI LEVEL INHERITANCE

- WHEN A CLASS EXTENDS **A CLASS** EXTEDS ANOTHER CLASS THEN IT IS CALLED AS MULI LEVEL INHERITANCE
- FOT EXAMPLE : CLASS C EXTENDS CLASS B AND CLASS B EXETENDS CLASS A THEN THIS TYPE OF INHERITANCE IS KNOWN AS MULTI LEVEL INHERITANCE

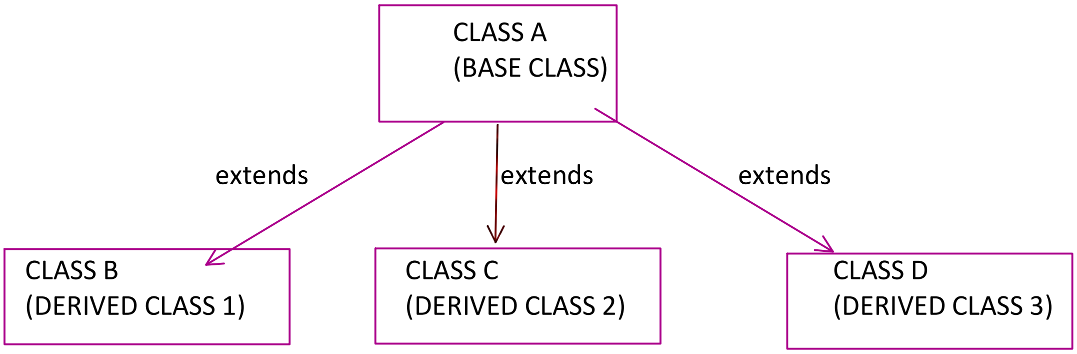


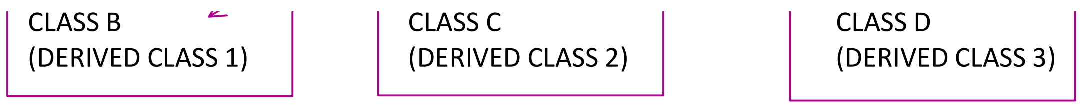


```
class Test1
{
        int a = 10
}

class Test2 extends Test1
{
        int b = 20;
}

class Test3 extends Test2
{
        public static void main(String[] args)
        {
                    Test3 a1 = new Test3();
                    System.out.println(a1.a);
                    System.out.println(a1.b);

        }
}
```

HIERARCHICAL INHERITANCE

- WHEN MORE THAN ONE CLASS INHERITS A SAME CLASS THEN THIS IS CALLED HIERARCHICAL INHERTIANCE.
- FROM EXAMPLE CLASS B , CLASS C CLASS D EXTENDS THE SAME CLASS A

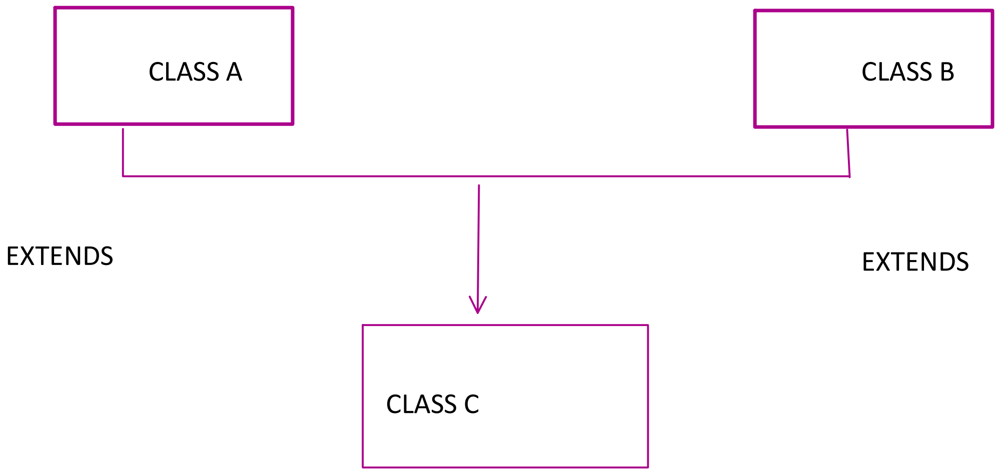




```
class Test1
{
	static int a = 23;
	String s = "BitsQ";

	public void m1()
	{
		System.out.println("in parent class meethod");
	}
}

class Test2 extends Test1
{
	static int b = 45;
	char ch = 'r';

	public void m2()
	{
		System.out.println("inside the Test2 class meethod");
	}
}

class Test3 extends Test1
{
	static int c = 21;
	double d = 9.16;

	public void m3()
	{
		System.out.println("in Test3 class meethod");
	}
	public static void main(String[] args)
	{
			System.out.println("the value of c is "+c);

			Test3 a1 = new Test3();
			a1.m3();

			System.out.println("the value of d is "+a1.d);
			a1.m1();
			System.out.println("the value of s is "+a1.s);
			System.out.println("the value of a is "+a);



	}
}
```

MULTIPLE INHERITANCE

- IN MULTIPLE INHERITACE , ONE CLASS WILL HAVE MULTIPLE PARENTS
- MULTIPLE INHERITANCE IS NOT SUPPORTED IN JAVA

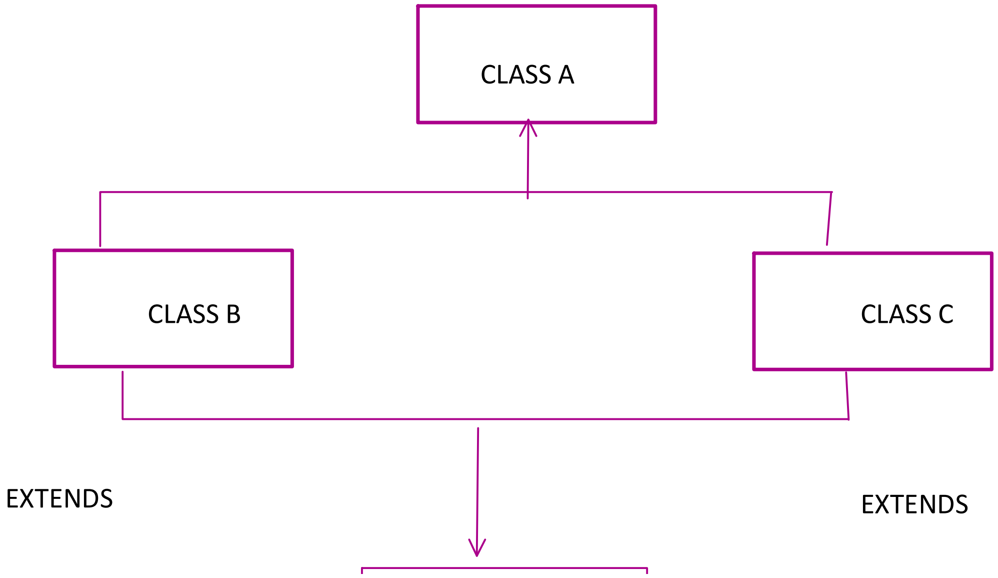


CONCLUSION

- EVERY CLASS IN JAVA IS SUB CLASS OF OBJECT CLASS.
- WE DON'T HAVE TO EXTEND EXPLICITLY FROM OBJECT CLASS, COMPILER AUTOMATICALLY EXTENDS FROM OBEJCT CLASS.
- OBJECT CLASS IS THE SUPER MOST CLASS IN JAVA
- IT WE DECLARE A CLASS USING FINAL KEYWORD, THEN IT CAN NOT HAVE ANY SUB CLASSES, IF WE TRY TO EXTEND FROM IT , WE WILL GET COMPILATION ERROR.

HYBRID INHERITANCE

- HYBRID INHERITANCE IS A COMBINATION OF HIERARCHICAL INHERTIANCE AND MULTIPLE INHERITANCE
- SINCE, MULTIPLE INHERITANCE IS NOT SUPPORTED IN JAVA, SO HYBRID INHERITANCE IS ALSO NOT SUPPORTED IN JAVA



CLASS D

```
final class Test1
{
        static int a = 23;
        String s = "BitsQ";

        public void m1()
        {
                System.out.println("in parent class method");
        }
}

class Test2 extends Test1
{
        public static void main(String[] args)
        {
                Test2 a1 = new Test2();
                System.out.println(a);
                System.out.println(a1.s);
                a1.m1();
        }
}
```

NOTE : CAN NOT BE DONE AS PARENT CLASS IS DECLARED AS FINAL CLASS