基于类的继承和多态。分析以下代码打印结果，并测试，弄清楚为什么？

class A {

public String show(D obj) {

return ("A and D");

}

public String show(A obj) {

return ("A and A");

}

}

class B extends A {

public String show(B obj) {

return ("B and B");

}

public String show(A obj) {

return ("B and A");

}

}

class C extends B {

}

class D extends B {

}

public class Test {

public static void main(String[] args) {

A a1 = new A();

A a2 = new B();

B b = new B();

C c = new C();

D d = new D();

System.out.println("1--" + a1.show(b));

System.out.println("2--" + a1.show(c));

System.out.println("3--" + a1.show(d));

System.out.println("4--" + a2.show(b));

System.out.println("5--" + a2.show(c));

System.out.println("6--" + a2.show(d));

System.out.println("7--" + b.show(b));

System.out.println("8--" + b.show(c));

System.out.println("9--" + b.show(d));

}

}

**class** A {

**public** String show(D obj) {

**return** ("A and D");

}

**public** String show(A obj) {

**return** ("A and A");

}

}

**class** B {

**public** String show(B obj) {

**return** ("B and B");

}

**public** String show(A obj) {

**return** ("B and A");

}

**public** String show(D obj) {

**return** ("A and D");

}

}

**class** C {

**public** String show(B obj) {

**return** ("B and B");

}

**public** String show(A obj) {

**return** ("B and A");

}

**public** String show(D obj) {

**return** ("A and D");

}

}

**class** D {

**public** String show(B obj) {

**return** ("B and B");

}

**public** String show(A obj) {

**return** ("B and A");

}

**public** String show(D obj) {

**return** ("A and D");

}

}

