Boolean b = new Boolean(“True”);

Boolean b1 = new Boolean(true);

System.out.Println(b.equals(b2));

Boolean b2 = new Boolean(“True”);

Boolean b3 = new Boolean(“TrUE”);

System.out.Println (b2.equals(b3));

Boolean b4 = new Boolean(“hey”);

Boolean b5 = new Boolean(“bye”);

System.out.Println (b4.equals(b5));

Ans: In all the above case, the output is true because Boolean wrapper class will take 2 diff args one as string and the other as Boolean itself. The strings are not case sensitive. Also, other than Boolean value as true or false, all other values will be considered as false and that is the case last is also true.

\*\*\*\*\*\*\*\*\*---------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------------\*\*\*\*\*\*\*\*\*\*\*\*

Is multiple inheritance allowed in java?

class A

{

    void methodOne()

    {

        System.out.println("From methodOfClassA");

    }

}

class B

{

    void methodOne()

    {

        System.out.println("From methodOfClassB");

    }

}

class C extends A, B

{

}

Answer:

In the above example, **Class A** also has **methodOne()** and **Class B** also has **methodOne()**. **Class C** is extending both the classes. So, both the methods with same name are inheriting to **Class C**. It causes confusion and ambiguity for which method to use. Therefore, to avoid this, multiple inheritance is not supported in java.

\*\*\*\*\*\*\*\*\*---------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------------------------\*\*\*\*\*\*\*\*\*\*\*\*

class A {

public void man() {

System.*out*.println("From A");

}

}

class B extends A {

@Override

void man() {

System.*out*.println("From B");

}

}

public class StaticBlock {

public static void main(String args[]) {

A obj = new A();

obj.man();

A obj1 = new B();

Obj1.man();

}}

Answer:

Cannot reduce the visibility of the inherited method from A

\*\*\*\*\*\*\*\*\*\*\*\*\*===================\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*======================\*\*\*\*\*\*\*\*

class A {

public static void man(){

System.*out*.println("A Man walks");

}

}

class B extends A{

public static void man(){

System.*out*.println("B Man walks");

}

}

public class Interview {

public static void main(String args[]) {

A obj = new A();

obj.*man*();

A obj1 = new B();

obj1.*man*();

}

}Answer: It will print A method since the method is static, it can’t be overridden and hence during compile time itself it is decided that it should print A method

A Man walks

A Man walks

\*\*\*\*\*\*\*\*\*\*\*\*\*---------------\*\*\*\*\*\*\*\*\*\*\*\*\*----------------\*\*\*\*\*\*\*\*\*

public class JavaStringsQuiz

{

    public static void main(String[] args)

    {

        String s = "1".repeat(5);

        System.out.println(s);

    }

}

Answer:

11111

--------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*------------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class StaticBlock {

public static void main(String args[]) {

String str = " Java\tConcept\tOf\tThe\tDay ".strip();

System.*out*.println(str);

}

}

Answer:

Remove all leading and trailing space in a string when we use the “.strip()” method

-------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*-----------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*

public class StaticBlock {

public static void main(String args[]) {

String s1 = "ONE";

s1.concat("TWO");

s1.concat("THREE");

System.*out*.println(s1);

}

}

Answer: One (Since string is immutable)

---------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*