

Q1. In the given problem, (s1==s2) is false as strings are immutable. On the contrary, (sb1==sb2) is true as stringbuffer objects are mutable. The result will be as:

(abcd abc false)

(abcd abcd true)

Q2. We have two overloaded methods named FlipRobo with different parameter types: one that takes a String and another that takes an Object. we then call FlipRobo(null) in the main method, passing null as the argument.

In your case, you are passing **null**, which can be treated as both a **String** or an **Object** since **null** is a reference that can point to any object.

However, when choosing between overloaded methods, Java prefers to select the most specific match.

In this case, the method with the **String** parameter is more specific than the one with the **Object** parameter, as **String** is a subclass of **Object**.

The output will be as :

String

Q3. The Third constructor is called, and it initializes the third class.

Before the third constructor executes, the constructor of the superclass (second) is called because third extends second So, second constructor is executed first.

Similarly, before the second constructor executes, the constructor of its superclass first is called because second extends first So, the first constructor is executed first.

The result will be as :

A

B

C

Q4. The calc method is called with the argument 2. Inside the calc method, the instance variable num is modified to 2 * 10, which results in the instance variable having a value of 20.

So, the program when run will print :

20

Q5. s1.indexOf(s2) attempts to find the index of the substring s2 ("Love") within s1 ("JavaLove").

However, the substring(4) operation didn't modify s1, and therefore, the result of this operation is still "JavaLove".

The indexOf method will find the first occurrence of "Love" in "JavaLove," which is at index 4.

System.out.println(foundAt) prints the value of foundAt, which is the index where "Love" is first found in s1. So, the output of your program will be:

4

Q6. the write method in the Writer class is static, so it is not overridden by subclasses, and it is not involved in method overriding and polymorphism.

The write method in the Author class is also static and prints "Writing book." However, in the main method, you are using the reference variable a, which is of type Author, so it will call the write method in the Author class.

The write method in the Programmer class is static and prints "Writing code." However, the reference variable a is of type Author, so this method won't be called in this context.

As a result, when you run the program, it will output:

Writing book

Q7. The output is "Not equal" because the == operator compares the references, and in this case, s1 and s2 reference two different String objects with the same content, making them not equal according to reference comparison. If we want to compare the content of the strings, we should use the equals method :

The result of the above method is:

Not equal

Q8. try Block:

The code within the try block attempts to execute the following:

It prints "First statement of try block."

It calculates the result of the division 45/3, which is 15, and stores it in the variable num.

It prints the value of num, which is 15.

catch Block:

The catch block catches any Exception that may occur during the execution of the try block. In this case, there isn't any exception raised, so this block will not be executed.

finally Block:

The finally block always gets executed, regardless of whether an exception is thrown or not. It prints "finally block."

After the finally block, the code proceeds to execute the following:

It prints "Main method."

The output will be as:

First statement of try block

15

finally block

Main method

Q9. At line 8, a static field a of type FlipRobo is declared and initialized with a new instance of the FlipRobo class. This means that as soon as the class is loaded, an instance of FlipRobo is created, and the constructor is called, so "constructor called" is printed.

In the main method at line 12, a local variable b of type FlipRobo is declared.

Then, a new instance of FlipRobo is created using b = new FlipRobo();. This means the constructor is called again, and "constructor called" is printed.

When we run the program, you will see the following output:

constructor called

constructor called

Q9. The first static block is executed before the constructor and prints "Static Block 1." Inside this block, num is set to 68, and mystr is set to "Block1."

The second static block is executed next and prints "Static Block 2." Inside this block, num is set to 98, and mystr is set to "Block2."

The constructor is then called, setting num to 100 and mystr to "Constructor."

Finally, the main method prints the values of num and mystr from the instance a. These values are 100 for num and "Constructor" for mystr.

So, the output of the program will be:

Static Block 1

Static Block 2

Value of num = 100

Value of mystr = Constructor