

1. Output:
abcd abc false
abcd abcd true

Explanation : In Java, String is immutable and string buffer is mutable.
So string s2 and s1 both pointing to the same string abc. And, after making the changes the string s1 points to abcd and s2 points to abc, hence false. While in string buffer, both sb1 and sb2 both point to the same object. Since string buffer are mutable, making changes in one string also make changes to the other string. So both string still pointing to the same object after making the changes to the object (here sb2).

2. Output: String

Explanation: In case of method overloading, the most specific method is chosen at compile time. As 'java.lang.String' is a more specific type than 'java.lang.Object'. In this case the method which takes 'String' as a parameter is chosen.

3. Output:

a
b
c

Explanation:
While creating a new object of 'Third' type, before calling the default constructor of Third class, the default constructor of super class is called i.e, Second class and then again before the default constructor of super class, default constructor of First class is called. And hence gives such output.

4. Output: 20

Explanation: Here the class instance variable name(num) is same as calc() method local variable name(num). So for referencing class instance variable from calc() method, this keyword is used. So in statement this.num = num * 10, num represents local variable of the method whose value is 2 and this.num represents class instance variable whose initial value is 100. Now in printNum() method, as it has no local variable whose name is same as class instance variable, so we can directly use num to reference instance variable, although this.num can be used.

5. Output: 4

Explanation: Append(String str) method, concatenate the str to s1. The substring(int index) method return the String from the given index to the end. But as there is no any String variable to store the returned string, so it will be destroyed. Now indexOf(String s2) method return the index of first occurrence of s2. So 4 is printed as s1="JavaLove".

6. Output: Writing book

Explanation: Since static methods can't be overridden, it doesn't matter which class object is created. As a is a Author referenced type, so always Author class method is called. If we remove write() method from Author class then Writer class method is called, as Author class extends Writer class.

7. Output: Not equal

Explanation: Since, s1 and s2 are two different objects the references are not the same, and the == operator compares object reference. So it prints "Not equal", to compare the actual characters in the string .equals() method must be used.

8. Output:

```
First statement of try block
15
finally block
Main method
```

Explanation:

Since there is no exception, the catch block is not called, but the finally block is always executed after a try block whether the exception is handled or not.

9. Output:

```
constructor called
constructor called
```

Explanation:

We know that static variables are called when a class loads and static variables are called only once. Now line 13 results to creation of object which in turn calls the constructor and "constructor called" is printed second time.

If in line 8 static variable would not have been used the object would have been called recursively infinitely leading to StackOverflow error.

10. Output:

```
Static Block 1
Static Block 2
Value of num = 100
Value of mystr = Constructor
```

Explanation:

Static block gets executed when the class is loaded in the memory. A class can have multiple Static blocks, which are executed in the same sequence in which they have been written into the program. As Static Methods can access class variables without using object of the class. Since constructor is called when a new instance is created so firstly the static blocks are called and after that the constructor is called. If we would have run the same program without using object, the constructor would not have been called.