
FULL STACK DEVELOPMENT – WORKSHEET-5- ANS

- Answers of Worksheet 5

Q1. Ans- 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.

Q2. Ans: String

Explanation: Object is a parent class and string is a child class. While resolving the overloading method, the compiler always uses the presidency for the child type argument. So, the compiler will call the method with the string argument.

Q3. Ans: 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.

Q4 Ans: 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.

Q5.Ans: 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".

Q6.Ans: Writing book

Explanation: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.

Q7. Ans: Not equal

Explanation: Here two String objects are being created namely s1 and s2. Both s1 and s2 refers to different objects. When one uses == operator for s1 and s2 comparison then the result is **not equal** as both have different addresses in memory.

Q8. Ans: First statement of try block
15
finally block
Main method

Explanation: In this case, default handling mechanism is followed. If finally block is present, it will be executed followed by default handling mechanism.

Q9. Ans: constructor called
constructor called

Explanation: Here the Constructor is a special type of method which is used to initialize the object. Every time an object is created using the new() keyword, at least one constructor is called. It calls a default constructor if there is no constructor available in the class. In such case, Java compiler provides a default constructor by default. Here **new FlipRobo() is used to call constructor.**

Q10. Output: Static Block 1
Static Block 2
Value of num = 100
Value of mystr = Constructor

Explanation: Here by this program static block 1 & 2 and static variables are executed in the order they are present in a program. When a variable is declared as static, then a single copy of the variable is created and shared among all objects at the class level. Static variables are, essentially, global variables. All instances of the class share the same static variable. We know static variable will get the memory only once, if any object changes the value of the static variable, it will retain its value.