WORKSHEET 5

1. The output will be:

abcd abc false

abcd abcd true

Because the string s1 holds the value “abc” and then the value inside s1 is put inside s2. Then letter ‘d’ is concatenated at end of s1. The print statement prints the value inside s1 and s2 and checks whether they are equal or not, which is obviously false.

Later, a stringbuffer sb1 is made holding the value “abc” . Another stringbuffer sb2 is made which holds the same value as sb1 i.e. both points to the same object. Then a letter ‘d’ is appended at the end of sb1. In java, stringbuffer is mutable, hence any changes occurred inside one string also makes changes inside the other. Hence, when the print statement is executed to check sb1 and sb2 equality, the result is true.

2. The output will be **String** because In case of method overloading, the most specific method is chosen and ‘java.lang.String’ is more specific type than ‘java.lang.Object’. This is why, this method taking ‘String’ as a parameter is chosen.

3. The output will be:

a

b

c

Because class Third has been extended from Second, which has been extended from class First. As a result, when an object for class Third is created, all the attributes of prior classes will be extended into the Third class.

4. The output will be **20** because in calc method which has taken one num parameter mutiplies a number which is passed in the main method by 2. Then printNum method displays that value.

5. The output will be **4** because the append method concatenates s1 to s2. The substring method returns the String from the input index to the end - Love. Now indexOf method returns the index of first occurrence of s2 in string s1=”JavaLove, which is 4.

6. The output will be **Writing Book,** as static methods can’t be overridden, but a is made the reference type for Author class and write method is called.

7. The output will be **Not Equal**, as both strings are made using new keyword, which allocates new memory location and address to both strings in the heap memory. Additionally, the “==” operator

checks the reference of the string, hence both strings, however holding the same data are rendered unequal.

8. The output will be:

First statement of try block

15

finally block

Main method

Because, as the whole code is inside the try block, it checks for any abnormality in the code, and the catch catches those exceptions. Here, the first print statement gets executed first, then division of 45/ 3 =15, happens which is stored inside a int data type, hence holds no decimal value. As the code runs normally, the statement inside catch block does not get executed. The finally keyword ensures that any statement inside it gets executed no matter if the exception was caught or not. Thus, “finally block” and “Main method” statements gets executed.

9. The output will be:

constructor called

constructor called

As in class FlipRobo, a constructor is made with a print statement, then a static object is created in the same class (line 8) with reference variable a. Later, inside the main method, another object for FlipRobo class is created with a reference variable b which calls the constructor again.

10. The output will be:

Static Block 1

Static Block 2

Value of num = 100

Value of mystr = Constructor

As static variables always get executed first even before the main method, we get such output. Then, inside the main method, an object for FlipRobo class is created which calls the constructor FlipRobo() holding the value of num = 100 and mystr = Constructor respectively, which gets printed out.