Worksheet 5

Q1. Output:

abcd abc false

abcd abcd true

Explanation:

In the program, the first part involves working with String objects. When s1 is initialized with "abc" and assigned to s2, both references point to the same memory location. However, when s1 is modified by concatenating "d", a new String object is created because String objects are immutable in Java. This results in s1 pointing to a new object "abcd", while s2 still points to "abc". Consequently, the comparison (s1 == s2) returns false since they now refer to different objects.

In the second part of the program, StringBuffer is used, which is mutable. Both sb1 and sb2 are references to the same StringBuffer object initialized with "abc". When sb1.append("d") is called, the original StringBuffer object is modified directly to "abcd". Since sb1 and sb2 still point to the same object, the comparison (sb1 == sb2) returns true.

Q2 Output: String

Explanation:

Method overloading is demonstrated with two versions of the FlipRobo method: one that accepts a String parameter and another that accepts an Object parameter. When the main method calls FlipRobo(null), the compiler has to decide which method to invoke. Since null can be assigned to any reference type, the compiler chooses the most specific method available. In this case, String is more specific than Object because String is a subclass of Object. Therefore, the FlipRobo(String s) method is invoked, and "String" is printed to the console.

Q3. Output:

a

b

c

Explanation:

In this program, we have a chain of inheritance involving three classes: First, Second, and Third. The Third class extends Second, which in turn extends First. When an object of the Third class is instantiated in the main method, the constructors of all the parent classes are invoked due to the nature of inheritance in Java.

The program execution goes like:

1. **Third c = new Third();**

- The Third class constructor is called.

2. **Constructor Chain:**

- Before the Third constructor body is executed, the Second constructor is called (since Third extends Second).
- Similarly, before the Second constructor body is executed, the First constructor is called (since Second extends First).

3. **Execution of Constructors:**

- The First class constructor executes and prints "a".
- Control returns to the Second class constructor, which then executes and prints "b".
- Finally, control returns to the Third class constructor, which executes and prints "c".

Q4. Output: 20

Explanation:

1. **Initialization:**

- A Calculator object obj is created. The instance variable num is initialized to 100.

2. **Method Call (obj.calc(2)):**

- The calc method is called with the argument 2.
- Inside the calc method, the parameter num (which is local to the method) takes the value 2.
- The expression this.num = num * 10; is executed. Here, this.num refers to the instance variable num of the object, while num refers to the method's parameter.
- The instance variable this.num is updated to 2 * 10 = 20.

3. **Method Call (obj.printNum()):**

- The printNum method is called, which prints the current value of the instance variable num, which is now 20.

Q5. Output: 4

Explanation:

In this program, a StringBuilder initialized with "Java" has "Love" appended to it, resulting in "JavaLove". The substring(4) method is called, but it doesn't alter the original StringBuilder content since the result is not stored. The indexOf(s2) method is then used to find the starting index of "Love" within "JavaLove", which is at index 4. Therefore, the program outputs 4.

Q6. Output: Writing book

Explanation:

Class Definitions:

- Writer has a static method write() that prints "Writing...".
- Author extends Writer and overrides the write() method to print "Writing book".
- Programmer extends Author and further overrides the write() method to print "Writing code".

In the main Method:

- An Author reference a is created and assigned a Programmer object. However, since write() is static, the method that gets called depends on the type of the reference (Author), not the type of the object (Programmer).

Method Call:

- a.write(); calls the static method write() from the Author class because the reference type is Author.

Q7. Output: Not Equal

Explanation:

Object Creation:

- String s1 = new String("FlipRobo"); creates a new String object s1 with the value "FlipRobo".
- String s2 = new String("FlipRobo"); creates another new String object s2 with the same value.

Comparison:

- The if (s1 == s2) statement compares the references of s1 and s2. Since new String("FlipRobo") creates distinct objects in memory, s1 and s2 are different objects, so the comparison s1 == s2 evaluates to false.

Output:

- Since s1 and s2 are not the same object, "Not equal" is printed.

Q8. Output:

First statement of try block

15

finally block

Main method

Explanation:

Try Block Execution:

- The first System.out.println("First statement of try block"); statement prints "First statement of try block".
- The expression int num = 45 / 3; performs integer division, resulting in 15, and System.out.println(num); prints 15.

Catch Block:

- The catch(Exception e) block is meant to handle any exceptions that might occur within the try block. However, no exceptions are thrown in this case, so this block is skipped.

Finally Block:

- The finally block is executed regardless of whether an exception was thrown or not. System.out.println("finally block"); prints "finally block".

Post-try-catch-finally Code:

- After the try-catch-finally block, System.out.println("Main method"); prints "Main method".

Q9. Output:

constructor called

constructor called

Explanation:

Static Variable Initialization:

- When the FlipRobo class is loaded, the static variable a is initialized. This involves executing the static initialization block or static variable initialization code.
- static FlipRobo a = new FlipRobo(); triggers the constructor call for FlipRobo, so "constructor called" is printed.

Instance Initialization:

- After the static initialization, the main method is executed.
- Within main, a new FlipRobo object is created with b = new FlipRobo();.
- This also triggers the constructor call, so "constructor called" is printed again.

Q10. Output:

Static Block 1

Static Block 2

Value of num = 100

Value of mystr = Constructor

Explanation:

1. **Static Block Execution:**
    - "Static Block 1" is printed first.
    - "Static Block 2" is printed next.
2. **Constructor Execution:**
    - When a FlipRobo object is created, the constructor sets num to 100 and mystr to "Constructor".
3. **Print Statements in main:**
    - System.out.println("Value of num = " + a.num); prints 100.

- System.out.println("Value of mystr = " + a.mystr); prints "Constructor".