Self Test: 8 correct / 19

My answer

Correct answer

~~Wrong answer~~

**Understand Fundamental Operators**

1. Given:

|  |
| --- |
| public class ArithmeticResultsOutput {  public static void main (String[] args) {  int i = 0;  int j = 0;  if (i++ == ++j) {  System.out.println("True: i=" + i + ", j=" + j);  } else {  System.out.println("False: i=" + i + ", j=" + j);  }  }  } |

What will be printed to standard output?

1. True: i=0, j=1
2. True: i=1, j=1
3. False: i=0, j=1
4. False: i=1, j=1
5. Which set of operators represents the complete set of valid Java assignment operators? (Note that shift operators include <<, >>, and >>>.)
6. %=, &=, \*=, $=, :=, /=, ^=, |=, +=, <<=, =, -=, >>=, >>>=
7. ~~%=, &=, \*=, /=, ^=, |=, +=, <<=, <<<=, =, -=, >>=, >>>=~~
8. %=, &=, \*=, /=, ^=, |=, +=, <<=, =, -=, >>=, >>>=
9. %=, &=, \*=, $=, /=, ^=, |=, +=, <<=, <<<=, =, -=, >>=, >>>=
10. Given the following Java code segment, what will be printed, considering the usage of the modulus operators?

|  |
| --- |
| System.out.print(49 % 26 % 5 % 1); |

1. 23
2. 3
3. 1
4. 0
5. Given:

|  |
| --- |
| public class BooleanResultsOutput {  public static void main (String[] args) {  boolean booleanValue1 = true;  boolean booleanValue2 = false;  System.out.print(!(booleanValue1 & !booleanValue2) + ", ");  System.out.print(!(booleanValue1 | !booleanValue2) + ", ");  System.out.print(!(booleanValue1 ^ !booleanValue2));  }  } |

What will be printed, considering the usage of the logical boolean operators?

1. false, false, true
2. false, true, true
3. true, false, true
4. true, true, true
5. Given:

|  |
| --- |
| public class ArithmeticResultsOutput {  public static void main (String[] args) {  int i1 = 100; int j1 = 200;  if ((i1 == 99) & (--j1 == 199)) {  System.out.print("Value1: " + (i1 + j1) + " ");  } else {  System.out.print("Value2: " + (i1 + j1) + " ");  }  int i2 = 100; int j2 = 200;  if ((i2 == 99) && (--j2 == 199)) {  System.out.print("Value1: " + (i2 + j2) + " ");  } else {  System.out.print("Value2: " + (i2 + j2) + " ");  }  int i3 = 100; int j3 = 200;  if ((i3 == 100) | (--j3 == 200)) {  System.out.print("Value1: " + (i3 + j3) + " ");  } else {  System.out.print("Value2: " + (i3 + j3) + " ");  }  int i4 = 100; int j4 = 200;  if ((i4 == 100) || (--j4 == 200)) {  System.out.print("Value1: " + (i4 + j4) + " ");  } else {  System.out.print("Value2: " + (i4 + j4) + " ");  }  }  } |

What will be printed to standard output?

1. Value2: 300 Value2: 300 Value1: 300 Value1: 300
2. Value2: 299 Value2: 300 Value1: 299 Value1: 300
3. Value1: 299 Value1: 300 Value2: 299 Value2: 300
4. Value1: 300 Value1: 299 Value2: 300 Value2: 299
5. Given the following code segment:

|  |
| --- |
| public void validatePrime() {  long p = 17496; // 'prime number' candidate  Double primeSquareRoot = Math.sqrt(p);  boolean isPrime = true;  for (long j = 2; j <= primeSquareRoot.longValue(); j++) {  if (p % j == 0) {  // Print divisors  System.out.println(j + "x" + p / j);  isPrime = false;  }  }  System.out.println("Prime number: " + isPrime);  } |

Which of the following is true? Hint: 17496 is not a prime number.

1. The code will not compile due to a syntactical error somewhere in the code.
2. The code will not compile since the expression (p % j == 0) should be written as ((p % j) == 0).
3. Divisors will be printed to standard output (for example, 2x8478, and so on), along with Prime number: false as the final output.
4. Divisors will be printed to standard output (for example, 2x8478, and so on), along with Prime number: 0 as the final output.
5. Given:

|  |
| --- |
| public class EqualityTests {  public static void main (String[] args) {  Integer value1 = new Integer("312");  Integer value2 = new Integer("312");  Object object1 = new Object();  Object object2 = new Object();  Object object3 = value1;  }  } |

Which expressions evaluate to true?

1. value1.equals(value2)
2. value1.equals(object1)
3. value1.equals(object3)
4. ~~object1.equals(object2)~~
5. Given:

|  |
| --- |
| System.out.print( true | false & true + "," );  System.out.println( false & true | true ); |

What will be printed to standard output?

1. ~~true, true~~
2. true, false
3. false, true
4. false, false
5. Compilation error
6. In the following code segment, what is printed to standard output?

|  |
| --- |
| int score = 10;  System.out.println("score: " + score++); |

1. 9
2. 10
3. ~~11~~
4. A compiler error will occur.
5. A runtime error will occur.
6. From highest precedence to lowest, which list of operators is ordered properly?
7. \*, +, &&, =
8. \*, &&, +, =
9. \*, =, &&, +
10. +, \*, &&, =

**Use String Objects and Their Methods**

1. Given:

|  |
| --- |
| System.out.print(3 + 3 + "3");  System.out.print(" and ");  System.out.println("3" + 3 + 3); |

What will be printed to standard output?

1. 333 and 333
2. 63 and 63
3. 333 and 63
4. 63 and 333
5. Consider the interface CharSequence that is a required argument in one of the replace method declarations:

|  |
| --- |
| public String replace(CharSequence target, CharSequence replacement) {  …  } |

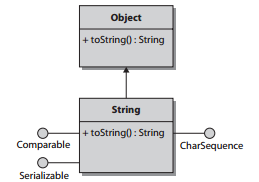
This CharSequence interface is a super interface to which concrete classes?

1. String
2. StringBoxer
3. StringBuffer
4. StringBuilder
5. Which statement is false about the toString method?
6. ~~The toString method is a method of the Object class.~~
7. ~~The toString method returns a string representation of the object.~~
8. The toString method must return the object’s state information in the form of a string.
9. The toString method is commonly
10. Which indexOf method declaration is invalid?
11. ~~indexOf(int ch)~~
12. ~~indexOf(int ch, int fromIndex)~~
13. indexOf(String str, int fromIndex)
14. indexOf(CharSequence str, int fromIndex)
15. Given:

|  |
| --- |
| String tenCharString = "AAAAAAAAAA";  System.out.println(tenCharString.replace("AAA", "LLL")); |

What is printed to the standard output?

1. AAAAAAAAAA
2. ~~LLLAAAAAAA~~
3. LLLLLLLLLA
4. LLLLLLLLLL
5. Consider the following illustration. Which statements, also represented in the illustration, are true?



1. The String class implements the Object interface.
2. The String class implements the Comparable, Serializable, and CharSequence interfaces.
3. The toString method overrides the toString method of the Object class, allowing the string object to return its own string.
4. The toString method is publicly accessible.

**Use StringBuilder Objects and Their Methods**

1. Which declaration of the StringBuilder class exists?
2. ~~public StringBuilder reverse (String str) {…}~~
3. public StringBuilder reverse (int index, String str) {…}
4. public StringBuilder reverse () {…}
5. All of the above

**Test Equality Between Strings and Other Objects**

1. Given:

|  |
| --- |
| String name1 = new String ("Benjamin");  StringBuilder name2 = new StringBuilder ("Benjamin");  System.out.println(name2.equals(name1)) |

Are the String and StringBuilder classes of comparable types? Select the correct statement.

1. ~~The String and StringBuilder classes are comparable types.~~
2. The String and StringBuilder classes are incomparable types.
3. Which append declaration does not exist in Java 8?
4. public StringBuilder append (short s) {…}
5. public StringBuilder append (int i) {…}
6. public StringBuilder append (long l) {…}
7. ~~public StringBuilder append (float f) {…}~~
8. public StringBuilder append (double d) {…}