



READ THIS NOW!

**Failure to read and follow the instructions below may result in severe penalties.
Failure to adhere to these directions will not constitute an excuse or defense.**

- Print your name in the space provided below.
- Print your name and ID number on the Opscan form; be sure to code your ID number on the Opscan form. Code **Form A** on the Opscan.
- Choose the single best answer for each question — some answers may be partially correct. If you mark more than one answer, it will be counted wrong.
- Unless a question involves determining whether given Java code is syntactically correct, assume that it is. The given code has been compiled and tested, except where there are deliberate errors. Unless a question specifically deals with Java import directives, you should assume the necessary library files have been imported.
- Note that questions about printed values disregard formatting completely.
- In questions/answers which require a distinction between integer and real values, integers will be represented without a decimal point, whereas real values will have a decimal point, [1054 (integer), 1054.0 (real)].
- When you have completed the test, sign the pledge at the bottom of this page and turn in the test and your Opscan.
- **This is a closed-book, closed-notes examination. No calculators or other electronic devices may be used during this examination. You may not discuss (in any form: written, verbal or electronic) the content of this examination with any student who has not taken it. You must return this test form when you complete the examination. Failure to adhere to any of these restrictions is an Honor Code violation.**
- There are 25 multiple-choice questions.
- Mark your answers on the test form, for future reference, and on the Opscan. The answers you mark on the Opscan form will be considered your official answers.

Do not start the test until instructed to do so!

Name _____ (print: Last name, First)

VT PID: _____ (print: campus email address)

Pledge: On my honor, I have neither given nor received unauthorized aid on this examination.

signature

1. What is the output of the following code:

```
String test2 = new String("test2");  
String t2 = new String("test2");  
if ( test2 == t2 )  
    System.out.println("SAME");  
else  
    System.out.println("DIFFERENT");
```

The equality operator compares the reference variables to determine if they are referring to the same object in memory.

- 1) SAME **2) DIFFERENT** 3) None of the above

2. In the Java language, what is the value of the expression: $4.8 + 5 / 2$

- 1) 4.9 2) 6 **3) 6.8**
4) 7 5) 7.3
6) The expression is not allowed in Java 7) None of these

3. In the Java language, what is the value of the expression: `double d3 = 17 / 4 ;`

- 1) 4 **2) 4.0** 3) 4.25 4) None of these

4. In the Java language, what is the value assigned to i4: `int i4 = -4 - 4 % 8 ;`

- 1) -1 2) 0 3) 8
4) -4 5) -4.5
6) The expression is not allowed in Java **7) None of these**

5. In the Java language, what is the value assigned to i5: `int i5 = (int) (3.5 + 2) ;`

- 1) 5** 2) 5.0 3) 5.5
4) 3 5) 3.7
6) The expression is not allowed in Java 7) None of these

6. Given the following incomplete Java code segment:

```
String e = "2.71828";  
double d = _____;
```

Choose the Java code below to replace the underlined blank above to convert the string into an equivalent double representation?

- 1) Double.valueOf(e).doubleValue()**
2) Double.doubleValue.valueOf(e)
3) double.valueOf(doubleValue(e))
4) double.DoubleValue.valueOf(e)
5) None of the above

2) $5 / 2$ is int division = 2 when added to the 4.8, which is a double, the 2 is converted to 2.0
3) the int divisions results in 4, which is converted to 4.0 for assignment to the double variable.
4) % is the remainder operator resulting in $-4 - 4$ which is -8
5) the 2 when added to the 3.5, which is a double, the 2 is converted to 2.0 resulting in 5.5 which is truncated to 5 for assignment to the int variable.

7. What output will the following program produce?

```
int Grade = 75, Category = 3;

if (Grade > 75)
    if (Category < 4)
        System.out.println("Bad");
else
    System.out.println("Good");
```

The else is matched with the previous unmatched if, which is the second if, thus the second if else is entirely under control of the first if whose condition is false.

- 1) Bad 2) Good 3) "Bad" 4) "Good"
- 5) both 1 and 2 6) both 3 and 4 **7) No Output** 8) none of these

8. What is the value printed for the variable Alpha if the following code is executed?

```
int Alpha = 2, X = 7;

if ( X % 2 == 1 )
    Alpha = Alpha + X;
X--;

if ( X % 3 == 0 )
    Alpha = Alpha + X;
X--;

if ( X / 3 == 0 )
    Alpha = Alpha + X;

System.out.println("Alpha = " + Alpha);
```

Performing the computations & substituting yields:
 if (7 % 2 == 1) { //which is true
 Alpha = Alpha + X; //Alpha = 9
 X--; //X = 6
 if (6 % 3 == 0) //which is true
 Alpha = Alpha + X; //Alpha = 9 + 6 = 15
 X--; //X = 5
 if (5 / 3 == 0) //which is false
 //nothing is executed

- 1) 13 2) 2 3) 9
- 4) 8 5) 16 6) 20
- 7) 23 **8) 15** 9) None of these

Given the following Java code segment:

```
int x, y, w = 0;
for (x = 0; x <= 4; x++)
{
    w = -1;
    for (y = 1; y < x; y++)
    {
        w = w + y;
    }
}
System.out.println( "w = " + w );
```

Execution Trace

<u>x</u>	<u>w</u>	<u>y</u>	
0	-1		
		1	
		<u>(No execution of inner for first time)</u>	
		since w is reset to -1 each time through the inner for only the last iteration of the loop needs to be traced.	
4	-1		
	0	1	
	2	2	
	5	3	

9. What is printed out for w when the above code is executed?

- 1) 10 2) 9 3) 8 4) 7 **5) 5**
- 6) 6 7) -1 8) none of the above

10. int A = 0, D = 5;
 if (R <= T)
 D = A - 1;
 else
 A = 1;

Hand execute each of the code fragments below and compare the resulting values of A & D to determine that 1 & 3 yield the same values as the fragment at the left.

Which of the following code fragments below is equivalent to the code above? "Equivalent" means that each code fragment would assign the same values to A and D as the code given above, no matter what the initial values for the variables R and T were.

- | | |
|--|---|
| <p>1) int A = 0, D = 5;
 if (R > T)
 A = 1;
 else
 D = A - 1;</p> <p>3) int A = 0, D = 5;
 if (R > T)
 A = 1;
 if (R <= T)
 D = A - 1;</p> <p>5) all of the above
8) 2 and 4 only</p> | <p>2) int A = 0, D = 5;
 if (R >= T)
 A = 1;
 else
 D = A - 1;</p> <p>4) int A = 0, D = 5;
 if (R < T)
 A = 1;
 if (R > T)
 D = A - 1;</p> <p>6) 1 and 2 only
9) none are equivalent</p> |
|--|---|

7) 1 and 3 only

For the next 3 questions, consider execution of the following switch statement:

```
int Passed = 15;
String sPass = kb.readLine(); //read line from the keyboard
Passed = Integer.parseInt(sPass);
switch (Passed) {
    case 0: Passed = 5;
           break;
    case 1:
    case 3: Passed = 7;
    case 5: Passed = 11;
           break;
    default: Passed = 0;
}
```

When Passed is 2 no case matches, so the default clause is executed.

When Passed is 3 the third case matches, but since it contains no break the next case clause statement also executes.

What would the value of Passed be after execution of this code if the value read for sPass were:

Passed	1	2	3	4	5	6
11. 2	0	5	7	11	15	none of the above
12. 3	0	5	7	11	15	none of the above
13. 5	0	5	7	11	15	none of the above

For the following 4 questions, suppose the (file) input buffered stream reader `infile`, (which has already been defined and instantiated), contains the following data (there's one tab character between columns and a newline character immediately after the last character on each line):

12	13	42	23	17	99
7	20	69	11	19	63
9	28	41	55	78	60
81	33	96	47	71	10

What is the value of each of the indicated variables after the execution of the following program segment?

```
String uno = "1", dos = "2", tres = "3", quattro = "4";
String dataLine = null;
StringTokenizer st = null;

dataLine = infile.readLine(); //sets dataLine = " 12 13 42 23 17 99"
st = new StringTokenizer(dataLine); //sets st = dataLine
uno = (String) st.nextElement(); // uno = 12
dos = (String) st.nextElement(); // dos = 13
dataLine = infile.readLine(); //sets dataLine = " 7 20 69 11 19 63"
st = new StringTokenizer(dataLine); //sets st = dataLine
tres = (String) st.nextElement(); // tres = 7
quattro = (String) st.nextElement(); // quattro = 20
dataLine = infile.readLine(); //sets dataLine but does NOT change st
// "\t" = tab; " " = space; "\r" = carriage return; "\n" = newline
dos = (String) st.nextToken("\t \r\n"); // dos = 69
tres = (String) st.nextToken("\t \r\n"); // tres = 11
tres = (String) st.nextToken("\t \r\n"); // tres = 19
dataLine = infile.readLine();
```

	1	2	3	4	5
14. uno	12	13	57	42	None of these
15. dos	7	28	69	13	None of these
16. tres	20	28	19	41	None of these
17. quattro	11	10	20	28	None of these

For the next four questions, assume the following variable declarations and initializations:

```
Vector v18, v19, v20;  
Enumeration e19, e20;  
String s19, s20;  
int i21;  
v19 = new Vector();  
v20 = new Vector();  
v18.addElement("item18");           //Line: 18  
v19.addElement("item19");  
v20.addElement("item20");  
s19 = (String) e19.nextElement();  //Line: 19  
e20 = v20.elements();  
s20 = (String) e20.nextElement();  //Line: 20  
i21 = e20.size();                  //Line: 21
```

18. The statement on Line: 18 *// Has not been set to a new Vector object (or had anything added to it)*
- 1) Contains a logic error. 2) Is logically correct.
19. The statement on Line: 19 *//the e19 enumeration has not been associated with any vector*
- 1) Contains a logic error. 2) Is logically correct.
20. The statement on Line: 20
- 1) Contains a logic error. 2) Is logically correct.
21. The statement on Line: 21 *//the size() method is a vector method, not an enumeration method*
- 1) Contains a logic error. 2) Is logically correct.
-

For the next four questions, consider the (incomplete) function:

```
public static double floorAvg(Vector List, int Floor) {
    int num = 0;
    Enumeration Lenum = List.elements();
    double Avg = _____; // line 1

    for (int Idx = 0; Idx < List.size(); Idx++) {
        Integer ival = (Integer) Lenum.nextElement();

        if (_____ ) { // line 2
            num++;
            Avg = Avg + ival.intValue();
        }
    }

    Avg = _____; // line 3
    return _____; // line 4
}
```

The function floorAvg() is intended to take two parameters, including a vector of integers, and a "floor" value, and compute the average of values in the vector that are greater than the "floor".

22. How should the blank initializing the variable Avg in **line 1** be filled?

- 1) 0.0
- 2) List.size()
- 3) (int) Lenum.nextElement()
- 4) It doesn't matter.
- 5) None of these

Avg should be initialized to zero since it is used to hold the sum of the values.

23. How should the blank in **line 2** be filled?

- 1) Idx > Floor
- 2) ((Integer) Lenum.nextElement()).intValue() > Floor
- 3) ival.intValue() > Floor
- 4) Idx == Size
- 5) None of these

The intValue() Integer class method returns the corresponding int value which must be checked against the floor value to determine if it is in the desired set.

24. How should the blank in **line 3** be filled?

- 1) Avg / Lenum.size()
- 2) Avg / Floor
- 3) Avg / num
- 4) Avg / List.size()
- 5) None of these

Avg holds the sum of desired values and num stores a count of the values > Floor.

25. How should the blank in **line 4** be filled?

- 1) num
- 2) Avg
- 3) List.elements()
- 4) Avg + Floor
- 5) None of these