

Instructions. This exam has 7 questions, worth 10 points each. You have 50 minutes.

Resources. You may reference your optional one-sided 8.5-by-11 handwritten "cheat sheet" during this exam. You may not use the textbook, your notes, or any electronic devices. You may not communicate with anyone except the course staff during this exam.

Discussing this exam. Due to travel for extracurriculars and sports, some of your peers will take this exam next week. Do not discuss its contents with anyone who has not taken it.

This paper. Do not remove this exam from the exam room. You may fill in this page now.

NAME: _____

NETID: _____

PRECEPT: _____

EXAM ROOM: _____

"I pledge my honor that I will not violate the Honor Code during this examination."

SIGNATURE: _____

What do the following Java expressions evaluate to? Specify the value, then the type.

If the expression does not compile or causes an exception at runtime, put an **X** in both boxes.

Expression	Value	Type
<code>3 / 2 / 1</code>	<input type="text"/>	<input type="text"/>
<code>'3' + "3" + 3</code>	<input type="text"/>	<input type="text"/>
<code>Integer.parseInt("3.33")</code>	<input type="text"/>	<input type="text"/>
<code>Integer.parseInt('333')</code>	<input type="text"/>	<input type="text"/>
<code>3 * 3 - 3 * 3</code>	<input type="text"/>	<input type="text"/>
<code>"3" * 3</code>	<input type="text"/>	<input type="text"/>
<code>33 % 333</code>	<input type="text"/>	<input type="text"/>
<code>Math.max(3, Math.min(3.3, 3))</code>	<input type="text"/>	<input type="text"/>
<code>Math.sqrt(Math.pow(3, 2))</code>	<input type="text"/>	<input type="text"/>
<code>(!!!!true && !!!!false) != true</code>	<input type="text"/>	<input type="text"/>

For each statement below, select the term that best matches the description.

You may use each term once, more than once, or not at all.

- | | | |
|--|--------------------------|-----------------------|
| 1. Data type used to compute the factorial of 20. | <input type="checkbox"/> | A. int |
| 2. Separates statements in Java. | <input type="checkbox"/> | B. double |
| 3. Indicates that a function is part of the API. | <input type="checkbox"/> | C. short |
| 4. Must be unique for each method in a Java file. | <input type="checkbox"/> | D. long |
| 5. Default value of elements in an array of references. | <input type="checkbox"/> | E. float |
| 6. Used to specify an element in an array. | <input type="checkbox"/> | F. curly braces |
| 7. Cause a program to leave a method immediately. | <input type="checkbox"/> | G. square brackets |
| 8. Data type of a variable used to store <code>Math.E</code> . | <input type="checkbox"/> | H. parentheses |
| 9. A method cannot return a value if declared this way. | <input type="checkbox"/> | I. for loop |
| 10. Defines variable scope. | <input type="checkbox"/> | J. while loop |
| | | K. statement |
| | | L. signature |
| | | M. semicolon |
| | | N. public |
| | | O. private |
| | | P. void |
| | | Q. null |
| | | R. default |
| | | S. static |
| | | T. break |
| | | U. return |
| | | V. true |
| | | W. false |
| | | X. StackOverflowError |
| | | Y. infinite loop |
| | | Z. compilation error |

Consider the following program.

```
public class Loopy {
    public static void main(String[] args) {
        double[] a = { 4.5, 3.5, 6.0, 20.0, 3.0 };
        int n = a.length;
        double value = Double.POSITIVE_INFINITY;
        for (int i = 0; i < n; i++) {
            for (int j = i + 1; j < n; j++) {
                double result = Math.abs(a[i] - a[j]);
                if (result < value) {
                    value = result;
                }
            }
        }
        System.out.println(value);
    }
}
```

What does this program print?

Now let's assume that we can change the values in the array based on user input.

Describe, in 15 words or less, what this program would print.

Determine whether each of the following lines of code, having to do with arrays, compile or not.

	compiles	does not compile
1. <code>int[] a = int[10];</code>	<input type="radio"/>	<input type="radio"/>
2. <code>double[] a = int[10];</code>	<input type="radio"/>	<input type="radio"/>
3. <code>int[10] a = new int[10];</code>	<input type="radio"/>	<input type="radio"/>
4. <code>int[] a = {3, 6, 9}; int b = a[3];</code>	<input type="radio"/>	<input type="radio"/>
5. <code>int[] a;</code>	<input type="radio"/>	<input type="radio"/>
6. <code>int[] a = {3};</code>	<input type="radio"/>	<input type="radio"/>
7. <code>int[][] a = {{9, 12, 15}, {18, 21, 24}};</code>	<input type="radio"/>	<input type="radio"/>
8. <code>int[][] a = new int[10]; a[0] = new int[10];</code>	<input type="radio"/>	<input type="radio"/>
9. <code>int[][] a = {{3}, {3, 3}, {3, 3, 3}};</code>	<input type="radio"/>	<input type="radio"/>
10. <code>int[] a = new int[3]; a = new int[3];</code>	<input type="radio"/>	<input type="radio"/>

What does this program print? Put your answers in the boxes on the right.

```
public class MethodActing {
    public static int methodA(int a) {
        a = 0;
        return a;
    }
    public static void methodB(int[] b) {
        for (int i = 0; i < b.length; i++)
            b[i] = 2 * b[i];
    }
    public static void methodC(int[] c) {
        c = new int[c.length];
    }
    public static void main(String[] args) {
        int a = 1;
        methodA(a);
        System.out.println(a);
        int[] b = { 3, 6, 9 };
        methodB(b);
        for (int i = 0; i < b.length; i++)
            System.out.print(b[i] + " ");
        System.out.println();
        int[] c = { 12, 15 };
        methodC(c);
        for (int i = 0; i < c.length; i++)
            System.out.print(c[i] + " ");
        System.out.println();
        b = c;
        methodB(b);
        for (int i = 0; i < b.length; i++)
            System.out.print(b[i] + " ");
        System.out.println();
    }
}
```

This program has four lines of output. List them, in order, one per box, below.

Consider the following program.

```
public class NumberCruncher {
    public static void main(String[] args) {
        int current = StdIn.readInt();
        int previous = current;
        while (!StdIn.isEmpty()) {
            current = StdIn.readInt();
            StdOut.print((previous + current) / 2 + " ");
            previous = current;
        }
        StdOut.println();
    }
}
```

The file `input.txt` contains the following:

```
3 6 9 12 15
```

What does this program print when the user executes the following commands?

```
% java-introcs NumberCruncher < input.txt
```

```
% java-introcs NumberCruncher < input.txt | java-introcs NumberCruncher
```

A palindrome is a word that reads the same forward and backward. The following program prints a special kind of palindrome.

```
public class Palindrome {
    public static String palindrome(int i) {
        if (i == 0) return "S";
        if (i == 1) return "T";
        return palindrome(i - 2) + palindrome(i - 1) + palindrome(i - 2);
    }
    public static void main(String[] args) {
        int N = Integer.parseInt(args[0]);
        System.out.println(palindrome(N));
    }
}
```

What does this program print when the user executes:

```
% java-introcs Palindrome 3
```

What does this program print when the user executes:

```
% java-introcs Palindrome 4
```

How many T's are printed when the user executes:

```
% java-introcs Palindrome 8
```