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.

SOLUTION

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.

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 / 2 / 1</td>
<td>1</td>
<td>int</td>
</tr>
<tr>
<td>'3' + &quot;3&quot; + 3</td>
<td>“333”</td>
<td>String</td>
</tr>
<tr>
<td>Integer.parseInt(&quot;3.33&quot;)</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Integer.parseInt('333')</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3 * 3 - 3 * 3</td>
<td>0</td>
<td>int</td>
</tr>
<tr>
<td>&quot;3&quot; * 3</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>33 % 333</td>
<td>33</td>
<td>int</td>
</tr>
<tr>
<td>Math.max(3, Math.min(3.3, 3))</td>
<td>3.0</td>
<td>double</td>
</tr>
<tr>
<td>Math.sqrt(Math.pow(3, 2))</td>
<td>3.0</td>
<td>double</td>
</tr>
<tr>
<td>(!!!!true &amp;&amp; !!!false) != true</td>
<td>true</td>
<td>boolean</td>
</tr>
<tr>
<td>Question</td>
<td>Terminology</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
<td>---</td>
</tr>
</tbody>
</table>
Consider the following program.

```java
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?

```
0.5
```

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.

The minimum absolute value difference between any two pairs of elements.
Determine whether each of the following lines of code, having to do with arrays, compile or not.

<table>
<thead>
<tr>
<th></th>
<th>compiles</th>
<th>does not compile</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>int[] a = int[10];</td>
<td>○</td>
</tr>
<tr>
<td>2.</td>
<td>double[] a = int[10];</td>
<td>○</td>
</tr>
<tr>
<td>3.</td>
<td>int[10] a = new int[10];</td>
<td>○</td>
</tr>
<tr>
<td>4.</td>
<td>int[] a = {3, 6, 9}; int b = a[3];</td>
<td>●</td>
</tr>
<tr>
<td>5.</td>
<td>int[] a;</td>
<td>●</td>
</tr>
<tr>
<td>6.</td>
<td>int[] a = {3};</td>
<td>●</td>
</tr>
<tr>
<td>7.</td>
<td>int[][] a = {{9, 12, 15}, {18, 21, 24}};</td>
<td>●</td>
</tr>
<tr>
<td>8.</td>
<td>int[][] a = new int[10]; a[0] = new int[10];</td>
<td>○</td>
</tr>
<tr>
<td>9.</td>
<td>int[][] a = {{3}, {3, 3}, {3, 3, 3}};</td>
<td>●</td>
</tr>
<tr>
<td>10.</td>
<td>int[] a = new int[3]; a = new int[3];</td>
<td>●</td>
</tr>
</tbody>
</table>
What does this program print? Put your answers in the boxes on the right.

```java
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.

1

6 12 18

12 15

24 30
Consider the following program.

```java
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?

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

```
4 7 10 13
```

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

```
5 8 11
```
A palindrome is a word that reads the same forward and backward. The following program prints a special kind of palindrome.

```java
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
TSTST
```

What does this program print when the user executes:

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

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

```
% java-introcs Palindrome 8
85
```