

Instructions. This exam has seven (7) questions worth a total of seventy (70) points. You have fifty (50) minutes. Write all answers **inside** the designated spaces. Write neatly and legibly.

This exam is preprocessed by computer. If you use pencil (and eraser), write darkly. Write all answers inside the designated rectangles. When asked to fill in circles, do so completely with a dark pencil: ●. (not ✓ or ✕) If you change your mind, you must erase completely and fill in another circle!

Resources. The exam is closed book, except that you are allowed to use a one-page reference sheet (8.5-by-11 paper, one side, in your own handwriting). No electronic devices are permitted.

Discussing this exam. Discussing the contents of this exam before solutions have been posted is a violation of the Honor Code.

This exam. Do not remove this exam paper from this room. Write your name, NetID, precept, and the room in which you are taking the exam in the space below. Also, write and sign the Honor Code pledge. You may enter this information now.

NAME: _____

NETID (not email alias): _____

PRECEPT: _____

EXAM ROOM: _____

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

SIGNATURE: _____

Give the value of each of the following Java expressions. To express your answer, write a Java literal of the appropriate type, such as 0 (for an int), 3.14 (for a double), false (for boolean), "tiger" (for a String), 'a' (for a char). In other words, a double value must include a decimal point (and not scientific notation), a String value must include quotes, a char value must include single quotes, and a boolean value must be lowercase. If an expression results in a compile-time or run-time error, write ERROR (only the word ERROR in all CAPS) for its value.

int x = 20; int y = 2; double z = 1.0;

	Java Expression	Value
a.	y / x	
b.	x * y * z	
c.	Double.parseDouble("" + z)	
d.	(double) (y / x)	
e.	(double) y / x	
f.	(x >= y) && !(x < y)	
g.	x / y * z	
h.	(char) ((int) 'a')	
i.	(int)"3.14159"	
j.	x * z / y	

In each box below, enter the single capital letter corresponding to the best-matching description of that part of the program. You may use each letter once, more than once, or not all.

```

public static double[][] random(int m, int n) {
    double[][] a = new double[m][n];
    for (int i = 0; i < m ; i++)
        for (int j = 0; j < n; j++)
            a[i][j] = StdRandom.uniform(0.0, 1.0);
    return a;
}

```

The diagram shows the following connections:

- A box around `double[][]` in the return type is connected to the top-most empty box.
- A box around `int n` in the parameter list is connected to the second empty box from the top.
- A box around `i < m` in the first for loop is connected to the third empty box from the top.
- A box around `StdRandom` in the assignment is connected to the fourth empty box from the top.
- A box around `return a;` is connected to the bottom-most empty box.

A - Transfers control to caller

B - Evaluates to a boolean value

C - Initializes a value

D - Calls a function

E - Defines an API

F - Argument declaration

G - Creates an array

H - Indicates a library name

I - A primitive type

J - An array index

K - Evaluates to an int value

L - A return type

Consider this incomplete program - see statement 5 below:

```

1  public class Q3 {
2      public static void main(String[] args) {
3          for (int i = 0; i < 3; i++) {
4              for (int j = 0; j < 3; j++) {
5                  if ( )
6                      System.out.print("X");
7                  else
8                      System.out.print("0");
9              }
10             System.out.println("");
11         }
12     }
13 }

```

Suppose you insert each expression below into the parentheses in line 5. What pattern will be printed? For each expression on the left, write the capital letter of the best-matching pattern from the right. You may use each letter once, more than once, or not at all.

$(i + j) > 4$



$i \neq j$

$i \% 2 == 0$

$i == 0 \ || \ j == 0$

$(i + j) \% 2 == 0$

A	XXX X00 X00	B	00X 00X XXX
C	XXX 000 XXX	D	XXX XXX XXX
E	0XX XOX XX0	F	XOX OXO XOX
G	000 000 000		

For each of the following situations, which require storing all the data (using an array) and which can be computed without using an array? Assume all input is read from standard input using `StdIn`. Fill in the corresponding circle.

	Requires an array	Does not require an array
a. Read the titles and duration (in seconds) of N songs and print the average duration of all songs.	<input type="radio"/>	<input type="radio"/>
b. Read the titles and duration (in seconds) of N songs and print the name of each song whose duration is greater than a given threshold.	<input type="radio"/>	<input type="radio"/>
c. Read the titles and duration (in seconds) of N songs and determine if they are in ascending order by duration.	<input type="radio"/>	<input type="radio"/>
d. Read the titles and duration (in seconds) of N songs and create a playlist of the N songs in reverse order of how the songs appear in the input.	<input type="radio"/>	<input type="radio"/>
e. Read the title and duration (in seconds) of N songs and generate a randomly ordered playlist of N songs such that each song is played exactly once.	<input type="radio"/>	<input type="radio"/>

Consider the following Java code fragment:

```

1      double[] a = ... // see below
2      int N = a.length;
3      int low = -1;
4      int high = -1;
5      // Double.POSITIVE_INFINITY is greater than any other double
6      double delta = Double.POSITIVE_INFINITY;
7      for (int i = 0; i < N; i++)
8          for (int j = i + 1; j < N; j++) {
9              double value = Math.abs(a[i] - a[j]);
10             if (value < delta) {
11                 delta = value;
12                 low = i;
13                 high = j;
14             }
15         }
16     }
17

```

For each initialization of `double[] a` (below), what are the values of `low` and `high` after the code completes:

	initialization	low	high
a.	<code>double[] a = {};</code>		
b.	<code>double[] a = {1.0};</code>		
c.	<code>double[] a = {2.0, 1.0};</code>		
d.	<code>double[] a = {8.0, 1.0, 3.0, 2.0, 10.0};</code>		

e. Fill in the circle corresponding to the **single** statement that is true for the code snippet above:

This code produces predictable results for arrays only containing positive doubles.

This code computes the indices of the first pair of array elements that have the smallest distance between the array elements.

This code computes the indices of the last pair of array elements that have the smallest distance between the array elements.

This code computes the indices of the first pair of array elements that have the greatest distance between the array elements.

Consider the following code.

```

1  public class Q6 {
2      public static void main(String[] args) {
3          int n = StdIn.readInt();
4          int[] a = new int[n];
5          for (int i = 0; i < n; i++)
6              a[n - 1 - i] = StdIn.readInt();
7
8          StdOut.print(n);
9          for (int i = 0; i < n; i++)
10             StdOut.print(" " + a[i]);
11         StdOut.println();
12     }
13 }

```

Suppose that the file data1 contains

9 1 2 3 4 5 6 7 8 9

and the file data2 contains

1 2 3 4 5

For each command, give the output that is printed on the terminal. Do not indicate whitespace in the output. If the command leads to no output printed on the terminal, write **NOTHING** (in all CAPS).

<code>java-introcs Q6 < data2</code>	
<code>java-introcs Q6 java-introcs Q6</code>	
<code>java-introcs Q6 < data1 java-introcs Q6</code>	
<code>java-introcs Q6 data1</code>	
<code>java-introcs Q6 < data2 > data3</code>	

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

```
1 public class Han {
2
3     public static String[] f(String[] a) {
4         String[] b = new String[a.length];
5         for (int i = 0; i < a.length; i++)
6             b[i] = a[i];
7         a[0] = "Kitty";
8         return b;
9     }
10
11     public static int g(int a) {
12         a = 1000;
13         return a / 2;
14     }
15
16     public static double h(double c, double d) {
17         c = 2 * c;
18         d = 2 * d;
19         System.out.println(c + d);
20         return ((c - d) * (c - d));
21     }
22
23     public static void main(String[] args) {
24         int a = 1;
25         String[] s = {
26             "Gen", "Lara Jean", "Chris",
27             "Peter", "Trevor"
28         };
29         int b = g(a);
30         String[] t = f(s);
31         double c = 2.0, d = 4.0;
32         double e = h(c, d);
33         System.out.println(a);
34         System.out.println(b);
35         System.out.println(s[0]);
36         System.out.println(t[0]);
37     }
38 }
```

When this program is executed, it prints five (5) lines. Show the contents of each line, in order, one per box, below. Do not use quotes (").