COS126 **Written Exam 1** **Fall** '21

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, <u>write and sign</u> 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: ______________________________

Question 1	**Data Types**	**10** points

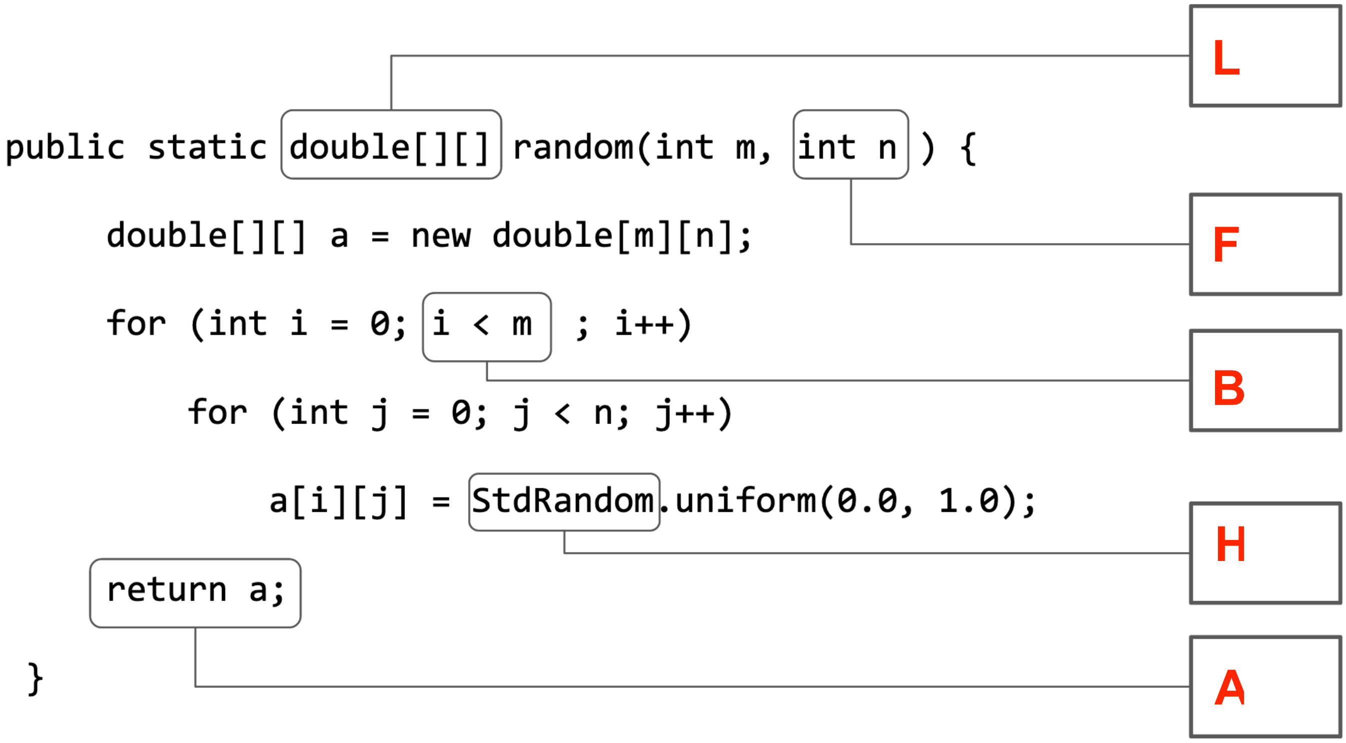
Give the value of each of the following Java expressions. To express your answer, write a Java literal of the appropriate type, such as `0` (for an `int`), `3.14` (for a `double`), `false` (for `boolean`), `"tiger"` (for a `String`), `'a'` (for a `char`). In other words, a `double` value must include 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`	0
b.	`x * y * z`	40.0
c.	`Double.parseDouble("" + z)`	1.0
d.	`(double) (y / x)`	0.0
e.	`(double) y / x`	.1
f.	`(x >= y) && !(x < y)`	true
g.	`x / y * z`	10.0
h.	`(char) ((int) 'a')`	'a'
i.	`(int)"3.14159"`	ERROR
j.	`x * z / y`	10.0

Question 2 **Java Syntax and Terminology** **10** points

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;
}
```

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

Question 3 **Conditionals and Loops** **10** points

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("O");
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.

G `(i + j) > 4`

E `i != j`

C `i % 2 == 0`

A `i == 0 || j == 0`

F `(i + j) % 2 == 0`

A **XXX** **X**OO **X**OO	B OO**X** OO**X** **XXX**
C **XXX** OOO **XXX**	D **XXX** **XXX** **XXX**
E O**XX** **X**O**X** **XX**O	F **X**O**X** O**X**O **X**O**X**
G OOO OOO OOO	

Question 4 **Arrays** **10** points

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.	○	●
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.	○	●
c.	Read the titles and duration (in seconds) of N songs and determine if they are in ascending order by duration.	○	●
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.	●	○
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.	●	○

Question 5 **Loops, Conditionals and Arrays** **10** points

Consider the following Java code fragment:

```java
double[] a = ... // see below
int N = a.length;
int low = -1;
int high = -1;
// Double.POSITIVE_INFINITY is greater than any other double
double delta = Double.POSITIVE_INFINITY;
for (int i = 0; i < N; i++)
  for (int j = i + 1; j < N; j++) {
     double value = Math.abs(a[i] - a[j]);
     if (value < delta) {
       delta = value;
       low = i;
       high = j;
     }
  }
```

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

	initialization	`low`	`high`
a.	`double[] a = {};`	-1	-1
b.	`double[] a = {1.0};`	-1	-1
c.	`double[] a = {2.0, 1.0};`	0	1
d.	`double[] a = {8.0, 1.0, 3.0, 2.0, 10.0};`	1	3

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. ○

Question 6 **StdIn, StdOut, Redirection, Piping** **10** points

Consider the following code.

```
public class Q6 {
    public static void main(String[] args) {
        int n = StdIn.readInt();
        int[] a = new int[n];
        for (int i = 0; i < n; i++)
            a[n - 1 - i] = StdIn.readInt();

        StdOut.print(n);
        for (int i = 0; i < n; i++)
            StdOut.print(" " + a[i]);
        StdOut.println();
    }
}
```

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).

`java-introcs Q6 < data2`	1 2
`java-introcs Q6 | java-introcs Q6`	NOTHING
`java-introcs Q6 < data1 | java-introcs Q6`	9 1 2 3 4 5 6 7 8 9
`java-introcs Q6 data1`	NOTHING
`java-introcs Q6 < data2 > data3`	NOTHING

Question 7 **Functions** **10** points

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

```
public class Han {

    public static String[] f(String[] a) {
        String[] b = new String[a.length];
        for (int i = 0; i < a.length; i++)
            b[i] = a[i];
        a[0] = "Kitty";
        return b;
    }

    public static int g(int a) {
        a = 1000;
        return a / 2;
    }

    public static double h(double c, double d) {
        c = 2 * c;
        d = 2 * d;
        System.out.println(c + d);
        return ((c - d) * (c - d));
    }

    public static void main(String[] args) {
        int a = 1;
        String[] s = {
         "Gen", "Lara Jean", "Chris",
         "Peter", "Trevor"
        };
        int b = g(a);
        String[] t = f(s);
        double c = 2.0, d = 4.0;
        double e = h(c, d);
        System.out.println(a);
        System.out.println(b);
        System.out.println(s[0]);
        System.out.println(t[0]);
    }
}
```

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 (").

12.0

1

500

Kitty

Gen