

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

| Expression | Value | Type |
|-------------------------------|--------------|----------------|
| "1" + 2 * 6 | "112" | String |
| 1 / 2 / 6 | 0 | int |
| Integer.parseInt(12) * 6 | X | X |
| (1 >= 26) (12 >= 6) | true | boolean |
| (int) 1.26 / (double) 10 | 0.1 | double |
| 1.2e6 % 1.2e5 | 0.0 | double |
| 1 <= 2 < 6 | X | X |
| Math.min(1.2e6, 1.2e7, 1.2e8) | X | X |
| true != false && true > false | X | X |
| !!!!!!false | false | boolean |

Select the term that best matches each statement below. You may use each term once, more than once, or not at all.

- | | | |
|---|----------|-----------------------|
| 1. Data type used to store large whole numbers precisely. | D | A. int |
| 2. Specifies a function's name, arguments, and return type. | L | B. double |
| 3. Used to override precedence of arithmetic operators. | H | C. short |
| 4. Specifies that a function does not return anything. | P | D. long |
| 5. Result of calling a recursive function with no base cases. | X | E. float |
| 6. Not required for a loop with a single statement. | F | F. curly braces |
| 7. Causes program to leave a loop immediately. | T | G. square brackets |
| 8. Causes program to leave a function immediately. | U | H. parentheses |
| 9. A type of error that is not a run-time or logic error. | Z | I. for loop |
| 10. Defines variable scope. | F | 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 |

Next to each piece of code, either mark the kind of error, or mark "no error" and write what is printed.

Assume that: `boolean A = false; boolean B = true; boolean C = false;`

1.

```
if (A) StdOut.println("A");
else  StdOut.println("B");
```

- compilation error
 runtime error or infinite loop
 no error; the output is: **B**

2.

```
if (A == true); StdOut.println("A");
```

- compilation error
 runtime error or infinite loop
 no error; the output is: **A**

3.

```
while (A == false) {
    if (B == false) A = false;
    if (C == false) B = false;
}
StdOut.println(B);
```

- compilation error
 runtime error or infinite loop
 no error; the output is: _____

NOTE: The variables A, B, and C are reset to their original values before running the code below.

4.

```
for (int i = 0; i < 100; i++) {
    boolean temp = A;
    A = B;
    B = temp;
}
StdOut.println(A);
```

- compilation error
 runtime error or infinite loop
 no error; the output is: **false**

5.

```
for (boolean D = true; C == D;)
    D = false;
StdOut.println(D);
```

- compilation error
 runtime error or infinite loop
 no error; the output is: _____

Determine whether each statement below is true or false. Assume no index is out of bounds. Write either "true" or "false" on each line.

For parts 8-11: `int[] a = new int[6]; int[] b = new int[6]; int[] c = a; c[0] = 3;`

1. Every element in a new array of integers is initialized to 0. **true**

2. Every element in a new array of booleans is initialized to true. **false**

3. N-1 is the last valid index into an array of size N. **true**

4. 0 is the index of the smallest element in an array of size N. **false**

5. N-3 is the index of the third-to-last element in an array of size N. **true**

6. `(int) (Math.random()*N)` is the index of a random element in an array of size N. **true**

7. You can create a new array without using the `new` keyword. **true**

8. Given the arrays `a`, `b`, and `c` as defined above, `a == b` evaluates to `true`. **false**

9. Given the arrays `a`, `b`, and `c` as defined above, `a == c` evaluates to `true`. **true**

10. Given the arrays `a`, `b`, and `c` as defined above, `a[b[0]]` evaluates to 0. **false**

11. Given the arrays `a`, `b`, and `c` as defined above, `a[b[c[0]]]` evaluates to 0. **false**

What do each of the following functions return? Fill in the tables below.

Assume that the arrays x , y , and z are defined as follows:

```
int[] x = new int[5]; int[] y = {1, 2, 6}; int[] z = {-10, 0, 10};
```

```
public static int f(int[] a) {
    int t = 0;
    int n = a.length;
    for (int i = 0; i < n; i++)
        t += a[i];
    return t;
}
```

| f(x) | f(y) | f(z) |
|----------|----------|----------|
| 0 | 9 | 0 |

```
public static int g(int[] a) {
    int n = a.length;
    for (int i = 0; i < n; i++)
        a[i] = a[n-i-1];
    return f(a);
}
```

| g(x) | g(y) | g(z) |
|----------|-----------|-----------|
| 0 | 14 | 20 |

NOTE: The arrays x , y , and z are reset to their original values before calling $h()$.

```
public static int h(int[] a) {
    int n = a.length;
    for (int i = 0; i < n; i++)
        a[i] = f(a);
    return f(a);
}
```

| h(x) | h(y) | h(z) |
|----------|-----------|-----------|
| 0 | 58 | 30 |

Consider the following program.

```
public class WordSmith {
    public static String choose(String str1, String str2) {
        // the .length() method returns the number of characters in a String
        if (str1.length() > str2.length())
            return str1;
        return str2;
    }
    public static void main(String[] args) {
        while (!StdIn.isEmpty()) {
            String word1 = StdIn.readString();
            String word2 = StdIn.readString();
            StdOut.print(choose(word1, word2) + " ");
        }
    }
}
```

The file `michelle.txt` contains the following quote:

```
It was clear to me that if I could get through Princeton at the top of my
class, I could do anything in the world. -- Michelle Obama
```

What is printed when the user runs the following commands?

```
% java-introcs WordSmith < michelle.txt
```

was clear that if could Princeton the top class, could anything the world. Michelle

(printed all on one line)

```
% java-introcs WordSmith < michelle.txt | java-introcs WordSmith
```

clear that Princeton top class, anything Michelle

This program produces the circles in the image below. Which circle is drawn first? Which is second? In each blank square, write the order the circles will appear. Use each number, 1 to 7, exactly once.

```
public class Circular {
    public static void horiz(int n, double x, double y, double r) {
        if (n == 0) return;
        vert(n - 1, x + r/2, y, r/2); // recur right
        vert(n - 1, x - r/2, y, r/2); // recur left
        StdDraw.circle(x, y, r);
    }
    public static void vert(int n, double x, double y, double r) {
        if (n == 0) return;
        horiz(n - 1, x, y + r/2, r/2); // recur up
        StdDraw.circle(x, y, r);
        horiz(n - 1, x, y - r/2, r/2); // recur down
    }
    public static void main(String[] args) {
        vert(3, .5, .5, .5);
    }
}
```

