COS 126 **General Computer Science** **Spring 2017**

Written Exam 1

This exam has 9 questions (including question 0) worth a total of 70 points. You have 50 minutes. Write all answers inside the designated spaces.

Policies. The exam is closed book, except that you are allowed to use a one page cheatsheet (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 from this room. Write your name, NetID, and the room in which you are taking the exam in the space below. Mark your precept number. Also, write and sign the Honor Code pledge. You may fill in this information now.

Name:

NetID:

Exam room:

Precept:

P01	P01A	P01B	P02	P02A	P03	P03A	P05	P06	P07	
○	○	○	○	○	○	○	○	○	○	
P08	P08A	P09	P11	P11A	P12	P13	P13A	P14	P14A	P15
○	○	○	○	○	○	○	○	○	○	○

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

Signature

0	1	2	3	4	5	6	7	8	total

0. **Miscellaneous. (1 point)**

   (a) Write your name, NetID, and the room in which you are taking the exam in the space provided on the front of this exam.

   (b) Mark your precept number on the front of this exam.

   (c) Write and sign the Honor Code pledge on the front of this exam.

1. **Java expressions. (10 points)**

   What is the value of each of the following Java expressions? To express your answer, write a Java literal of the appropriate type, such as `0`, `0.0`, `false`, or `"0"`. If an expression results in a compile-time or run-time error, write `ERROR` for its value. Assume that the variables `x`, `y`, and `z` have been initialized as follows:

```
int x = 2;
int y = 3;
int z = 4;
```

Java expression	*value*
`x`	2
`z + y / y + x`	
`x * y / z * z`	
`(z < x) && (z < y) || (x < y < z)`	
`(z - y - 1.0) / (z - x - 2.0)`	
`(double) (y / x)`	

2. **Java basics. (8 points)**

   Suppose that the variables `a`, `b`, `c` are of type `int`.

   (a) Complete the following code fragment so that, after the last statement, the variable `max` contains the largest of the three values.

   In each blank, write one of `a`*,* `b`*,* `c`*, or* `max`*. You may use each variable name once, more than once, or not at all.*

```
int max  =  a;

if ( ______  >  ______ )  ______  =  ______ ;

if ( ______  >  ______ )  ______  =  ______ ;
```

   (b) Complete the following code fragment to *rotate* the variables `a`, `b`, and `c`: after the last statement, `b` should store the old value of `a`; `c` should store the old value of `b`; and `a` should store the old value of `c`.

   In each blank, write one of `a`*,* `b`*,* `c`*, or* `temp`*. You may use each variable name once, more than once, or not at all.*

```
int temp  =  a;

______  =  ______ ;

______  =  ______ ;

______  =  ______ ;
```

3. **Loops and debugging. (9 points)**

What is the value of the variable `count` immediately after executing each of the following code fragments?

For each code fragment at left, choose the best-matching letter at right. You may use each letter once, more than once, or not at all.

______

```
int n = 10;
int count = 0;
for (int i = 0; i < n; i++) {
    count++;
    for (int j = 0; j < n; j++) {
        count++;
    }
}
```

______

```
int n = 10;
int count = 0;
for (int i = 0; i <= n; i++)
    count++;
    for (int j = 0; j <= n; j++)
        count++;
```

______

```
int n = 10;
int count = 0;
int i = 1;
while (i <= n) {
    count++;
    int j = 0;
    while (j <= n) {
        count++;
        j++;
    }
}
```

A. 0

B. 1

C. 2

D. 10

E. 11

F. 20

G. 22

H. 40

I. 100

J. 110

K. 120

L. 132

M. 220

N. 242

O. *infinite loop*

P. *run-time error*

4. **Properties of Java. (9 points)**

(a) Which of the following are features of Java types and variables? Mark each statement as either *true* or *false*.

true	*false*	
○	○	You must declare each variable to have a specific type, but you can change its type with a subsequent declaration statement.
○	○	You will get a compile-time error if you attempt to assign any value of type `double` to any variable of type `int`.
○	○	You will get a compile-time error if you use any variable in an expression before it has been declared.
○	○	You will get a run-time error if you attempt to use any variable in an expression before it has been initialized.
○	○	You will get a compile-time error if one operand of the multiplication operator (`*`) is of type `int` and the other is of type `double`.

(b) Which of the following are advantages of using `StdIn` instead of command-line arguments? Mark each statement as either *advantage* or *not advantage*.

advantage	*not advantage*	
○	○	You can enter input data *while* your program is executing.
○	○	You can execute your program using different input data without having to recompile it.
○	○	Your program can process the input data in reverse order.
○	○	Your program can process huge amounts of input data—more than can be stored in memory at any one time.

5. **Arrays. (9 points)**

Consider the following Java code fragment.

```
int[] x = { 1, 2, 3 };
int[] y = { 4, 5, 6 };
int n = x.length;

// Location 1.

for (int i = 0; i < n; i++) {

    // Location 2.

    for (int j = 0; j <= i; j++) {

        // Location 3.

        sum += x[j] * y[i-j];
    }
    StdOut.print(sum + " ");
}
```

Suppose that the statement `int sum = 0;` is put into one of the three designated locations above. What will the resulting code fragment do?

	compile-time error	*run-time error*	*print the following to standard output*
Location 1	○	○	
Location 2	○	○	
Location 3	○	○	

6. **Functions and booleans. (8 points)**

   The `minority()` function takes three boolean arguments and returns `true` if at most one of its arguments is `true`; otherwise, it returns `false`.

   Complete two *implementations of* `minority()` *by filling in the letter of one of the expressions below in each provided space. You may use a letter once, more than once, or not at all. No other code is allowed.*

```
A. true        C.  x        G. x && y        K. count--
B. false       D.  y        H. x && z        L. count++
               E.  z        I. x || y
               F. !z        J. y || z
```

(a)

```
public static boolean minority(boolean x, boolean y, boolean z) {

    int count = 0;

    if (_______) _______;

    if (_______) _______;

    if (_______) _______;

    return count <= 1;
}
```

(b)

```
public static boolean minority(boolean x, boolean y, boolean z) {

    if      (_______) return _______;

    else if (_______) return _______;

    else              return _______;
}
```

7. **Functions and arrays. (8 points)**

   Consider the following three Java functions.

```
public static int halve1(int x) {
    x = x / 2;
    return x;
}

public static void halve2(int[] a) {
    int n = a.length;
    for (int i = 0; i < n; i++) {
        halve1(a[i]);
        a[i] = halve1(a[i]);
    }
}

public static void halve3(int[] a) {
    int n = a.length;
    int[] b = new int[n/2];
    for (int i = 0; i < n/2; i++)
        b[i] = a[i];
    a = b;
}
```

(a) What are the contents of the array `a[]` after executing the following three lines of code? *Write your answer in the box below.*

```
int[] a = { 16, 32, 48, 64 };
halve2(a);
halve2(a);
```

(b) What are the contents of the array `a[]` after executing the following three lines of code? *Write your answer in the box below.*

```
int[] a = { 16, 32, 48, 64 };
halve3(a);
halve3(a);
```

8. **Recursion. (8 points)**

   Consider the following recursive function:

```
public static String f(int n) {
    if (n <= 0) return "0";
    if (n == 1) return "1";
    String first = f(n/2);
    String second = f(n - n/2);
    return first + " " + n + " " + second;
}
```

(a) Complete the following table by filling in the values of `f(2)` and `f(4)`.

n	f(n)
0	"0"
1	"1"
2	
4	

(b) Mark each of the following statements as either *true* or *false*.

true	*false*	
○	○	`f(8)` returns a string containing 15 integers, separated by whitespace.
○	○	If `n` is a power of 2, then `f(n)` returns a string containing $2n - 1$ integers, separated by whitespace.
○	○	If `n` is even, then `f(n)` returns a *palindromic* string—the string is equal to its reverse.
○	○	Suppose that the last statement is replaced by `return second + " " + n + " " + first;` Then, for every integer argument `n`, the modified function returns the *reverse* of the string returned by `f(n)`.

This page is provided as scratch paper. If you tear it out, write your name, NetID, and precept number in the space provided and return it inside your exam.

Name: ______________________ **NetID:** ____________ **Precept:** _______