There are seven questions on this exam, weighted as indicated at the top of each page. The questions are not in order of difficulty. If a question seems difficult to you, skip it and come back to it.

**Policies.** The exam is closed book, though you are allowed to use a single-page two-sided hand-written cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided. You have 50 minutes to complete the test. **This exam is preprocessed by computer.** If you use pencil (and eraser), write darkly. Write all answers inside the designated rectangles. Do not write on corner marks.

**This page.** *Print your name, NetID; fill in your precept on this page; write and sign the Honor Code pledge.*

**Discussing this exam.** As you know, discussing the contents of this exam before solutions have been posted is a serious violation of the Honor Code.

“I pledge my honor that I have not violated the Honor Code during this examination.”
Q1. Java Basic (10 points).

A. Assume that variables x, y, and z have been initialized as follows:

```java
int x = 111;
int y = 222;
int z = 333;
```

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, 0.0, false, or "0". If an expression results in a compile-time or run-time error, write ERROR for its value.

<table>
<thead>
<tr>
<th>Java expression</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>111</td>
</tr>
<tr>
<td>x - 2.0 * y + z</td>
<td></td>
</tr>
<tr>
<td>x / (z - x - y)</td>
<td></td>
</tr>
<tr>
<td>Math.sqrt(x / (x + z))</td>
<td></td>
</tr>
<tr>
<td>x + &quot;222&quot; + (y + z)</td>
<td></td>
</tr>
<tr>
<td>(x &lt;= y &lt;= z)</td>
<td></td>
</tr>
<tr>
<td>!(x &lt;= 2<em>y) &amp;&amp; (y &lt;= 2</em>x)</td>
<td></td>
</tr>
</tbody>
</table>
B. Suppose that the variables \(a\), \(b\), and \(c\) are initialized as follows:

\[
\begin{align*}
\text{int } a &= 11111; \\
\text{int } b &= 22222; \\
\text{int } c &= 0;
\end{align*}
\]

And consider the following statements.

1. \(a = b;\)
2. \(a = c;\)
3. \(c = a;\)
4. \(c = b;\)
5. \(b = a;\)
6. \(b = c;\)

Thus a sequence written \(1 3 5\) refers to the following sequence of statements:

\[
\begin{align*}
a &= b; \\
c &= a; \\
b &= c;
\end{align*}
\]

Which of the following sequences of statements will swap the values of \(a\) and \(b\), i.e., leave the value \(22222\) in \(a\) and \(11111\) in \(b\)? Mark all that apply.

\[
\begin{array}{cccc}
1 & 3 & 5 & \square \\
2 & 3 & 5 & \square \\
3 & 1 & 5 & \square \\
4 & 1 & 5 & \square
\end{array}
\]

\[
\begin{array}{cccc}
1 & 3 & 6 & \square \\
2 & 3 & 6 & \square \\
3 & 1 & 6 & \square \\
4 & 1 & 6 & \square
\end{array}
\]

\[
\begin{array}{cccc}
1 & 4 & 5 & \square \\
2 & 4 & 5 & \square \\
3 & 2 & 5 & \square \\
4 & 2 & 5 & \square
\end{array}
\]

\[
\begin{array}{cccc}
1 & 4 & 6 & \square \\
2 & 4 & 6 & \square \\
3 & 2 & 6 & \square \\
4 & 2 & 6 & \square
\end{array}
\]

\[
\begin{array}{cccc}
1 & 5 & 3 & \square \\
2 & 5 & 3 & \square \\
3 & 5 & 1 & \square \\
4 & 5 & 1 & \square
\end{array}
\]

\[
\begin{array}{cccc}
1 & 5 & 4 & \square \\
2 & 5 & 4 & \square \\
3 & 5 & 2 & \square \\
4 & 5 & 2 & \square
\end{array}
\]

\[
\begin{array}{cccc}
1 & 6 & 3 & \square \\
2 & 6 & 3 & \square \\
3 & 6 & 1 & \square \\
4 & 6 & 1 & \square
\end{array}
\]

\[
\begin{array}{cccc}
1 & 6 & 4 & \square \\
2 & 6 & 4 & \square \\
3 & 6 & 2 & \square \\
4 & 6 & 2 & \square
\end{array}
\]
Q2. Properties of Arrays and Functions (10 points).

A. Which of the following statements are true for Java arrays? Assume that the arrays are one-dimensional and integer-valued.

<table>
<thead>
<tr>
<th>Statement</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>The statement <code>int[] a = new int[n];</code> declares and creates a new array of length n, with each element initialized to 0.</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>You can refer to the first element in an array <code>a[]</code> with <code>a[0]</code> and the last element with <code>a[a.length]</code>.</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Each element <code>a[i]</code> in an array <code>a[]</code> can be used like an ordinary variable of type <code>int</code>: as a term in an expression or as the left-hand side in an assignment statement.</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>If <code>a[]</code> and <code>b[]</code> are arrays of the same length, then the expression <code>a + b</code> evaluates to a new array whose elements are the sum of the corresponding elements in <code>a[]</code> and <code>b[]</code>.</td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>The values of the elements in an array are stored consecutively in the computer’s memory.</td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>

B. Which of the following statements are true for Java functions (static methods)?

<table>
<thead>
<tr>
<th>Statement</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>A .java file can contain the definition of more than one function, but each function must have a different name.</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>Two different functions in the same .java file can declare local variables with the same name.</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>A function must contain exactly one <code>return</code> statement (unless its return type is <code>void</code>).</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>A function may either return a value to the caller or produce a side effect (such as consuming input or producing output), but cannot do both.</td>
<td></td>
<td>O</td>
</tr>
<tr>
<td>The scope of a local variable declared within a function is limited to that function.</td>
<td></td>
<td>O</td>
</tr>
</tbody>
</table>
Q3. Loops and Conditionals (10 points).

A. Consider the following Java code fragment:

```java
int N = 100;
for (int i = 0; i < N; i++) {
    for (int j = 0; (j < N) && (j != i); j++) {
        System.out.println(i + "-" + j);
    }
}
```

Which of the following will appear on standard output? Mark all apply.

![Checkboxes for 0-0, 0-1, 1-0, 8-8, 2-98, 98-2, 100-99]

B. Consider the following Java code fragment. Assume that x, y, and z are variables of type int and initialized to 10, 5, and 0, respectively.

```java
if (x >= y) {
    System.out.println("A");
    if (x >= z) {
        System.out.println("B");
    } else {
        System.out.println("C");
    }
} else {
    System.out.println("D");
}
if ((x < y) && (x < z)) {
    System.out.println("D");
} else {
    System.out.println("E");
    if (y >= z) {
        System.out.println("F");
    }
}
```

Which of the following will appear on standard output? Mark all that apply.

![Checkboxes for A, B, C, D, E, F, G]
Q4. Arrays (10 points).

A. Consider the following Java code fragment:

```java
int[] a = { 1, 6, 5, 3, 0, 2, 4 };
int n = a.length;

int[] b = new int[n];
for (int i = 0; i < n; i++)
    b[a[i]] = i;

int[] c = new int[n];
for (int i = 0; i < n; i++)
    c[i] = a[b[i]];
```

What are the values of the elements in the arrays `b[]` and `c[]` after the above code fragment is executed? Write your answers in the space below.

<table>
<thead>
<tr>
<th>a[]</th>
<th>1</th>
<th>6</th>
<th>5</th>
<th>3</th>
<th>0</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>b[]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>c[]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Q5. Standard Input, Standard Output, and Redirection (10 points).

Consider the following Java class contained in Mystery.java, and compiled:

```java
public class Mystery {
    public static void main(String[] args) {
        int current = -1;
        int count = 0;
        while (!StdIn.isEmpty()) {
            int x = StdIn.readInt();
            if (x != current) {
                if (count > 0) StdOut.print(count);
                current = x;
                count = 0;
            }
            count++;
        }
    }
}
```

A. Assume the contents of the file input1.txt are given below (note: the line starting with % is what you type on the command line prompt / the terminal; what follows is the actual content of the file input1.txt).

```bash
% more input1.txt
0 0 0 0 1 1 1 1
```

Suppose that you execute the following command. What is printed on the standard output? Write your answer in the space provided.

```bash
% java-introcs Mystery < input1.txt
```

B. Repeat the previous question, but with the following input file.

```bash
% more input2.txt
0 0 0 0 1 1 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1 0 1 1 1
```

```bash
% java-introcs Mystery < input2.txt
```
Q6. Functions (10 points).

The `majority()` function takes three boolean arguments and returns `true` if two (or more) of the arguments are `true`; and `false` otherwise. Complete two implementations of `majority()` below by filling in the letter of one of the expressions below in each provided space. You may use each letter once, more than once, or not at all (but you may not use any other code).

A.  

```java
public static boolean majority(boolean x, boolean y, boolean z) {
    int count = 0;
    if (_____ _____);
    if (_____ _____);
    if (_____ _____);
    return _____;
}
```

B.  

```java
public static boolean majority(boolean x, boolean y, boolean z) {
    if (_____ return _____;
    else if (_____ return _____;
    else if (_____ return _____;
    else return _____;
}
```

A.  

- **false**  
- **true**  
- **x**  
- **y**  
- **z**  
- **G.** `x && y`  
- **K.** `x || y`  
- **O.** `count++`

B.  

- **H.** `x && z`  
- **L.** `x || z`  
- **P.** `count--`

C.  

- **I.** `y && z`  
- **M.** `y || z`  
- **Q.** `count`

D.  

- **J.** `x && y && z`  
- **N.** `x || y || z`  
- **R.** `count <= 1`

E.  

- **G.** `x && y`  
- **K.** `x || y`  
- **O.** `count++`

F.  

- **H.** `x && z`  
- **L.** `x || z`  
- **P.** `count--`

G.  

- **I.** `y && z`  
- **M.** `y || z`  
- **Q.** `count`

H.  

- **J.** `x && y && z`  
- **N.** `x || y || z`  
- **R.** `count <= 1`

I.  

- **G.** `x && y`  
- **K.** `x || y`  
- **O.** `count++`

J.  

- **H.** `x && z`  
- **L.** `x || z`  
- **P.** `count--`

K.  

- **I.** `y && z`  
- **M.** `y || z`  
- **Q.** `count`

L.  

- **J.** `x && y && z`  
- **N.** `x || y || z`  
- **R.** `count <= 1`

M.  

- **G.** `x && y`  
- **K.** `x || y`  
- **O.** `count++`

N.  

- **H.** `x && z`  
- **L.** `x || z`  
- **P.** `count--`

O.  

- **I.** `y && z`  
- **M.** `y || z`  
- **Q.** `count`

P.  

- **J.** `x && y && z`  
- **N.** `x || y || z`  
- **R.** `count <= 1`

Q.  

- **G.** `x && y`  
- **K.** `x || y`  
- **O.** `count++`

H.  

- **H.** `x && z`  
- **L.** `x || z`  
- **P.** `count--`

I.  

- **I.** `y && z`  
- **M.** `y || z`  
- **Q.** `count`

J.  

- **J.** `x && y && z`  
- **N.** `x || y || z`  
- **R.** `count <= 1`
Q7. Recursive Graphics (10 points).

Design a recursive function with the signature:

```java
public static void draw(int n, double x, double y, double length)
```

so that the call `draw(5, 0.5, 0.5, 0.5)` produces the following *intermediate* result after drawing the 214th shaded square.

The six statements in the function body are given below, but not necessarily in order.

1. `if (n == 0) return;`
2. `drawShadedSquare(x, y, length);`
3. `draw(n-1, x - length/2, y + length/2, length/2.2);` // upper left
4. `draw(n-1, x + length/2, y + length/2, length/2.2);` // upper right
5. `draw(n-1, x - length/2, y - length/2, length/2.2);` // lower left
6. `draw(n-1, x + length/2, y - length/2, length/2.2);` // lower right

The helper function `drawShadedSquare()` draws a gray square of the specified side length, outlined in black, and centered at `(x, y)`. Which of the following must be true for any possible ordering of the six statements that produces the intermediate result shown above? Mark all that apply.

- [ ] Statement 1 appears first.
- [ ] Statement 2 appears after statement 5.
- [ ] Statement 2 appears before statements 3, 4, 5, and 6.
- [ ] Statements 3 and 6 appear before statements 4 and 5.
- [ ] If statements 3 and 6 are swapped, the function will still produce the same intermediate result.