

Name: \_\_\_\_\_

NetID: \_\_\_\_\_

Answer all questions in the space provided. Write clearly and legibly, you will not get credit for illegible or incomprehensible answers. Print your name at the top of every page.

This is a closed book exam. However, each student is allowed to bring one page of notes to the exam. Also, you are permitted the use of a “dumb” calculator to perform basic arithmetic.

Question:	1	2	3	4	5	6	7	8	Total
Points:	10	8	26	14	12	14	6	10	100
Score:									

## 1. True or false questions

- (a) An array can be passed as a parameter. (1)  
 (a) \_\_\_\_\_
- (b) An object can be returned by a method. (1)  
 (b) \_\_\_\_\_
- (c) It is legal to have more than one constructor in a given class. (1)  
 (c) \_\_\_\_\_
- (d) All the elements of an array must be the same type. (1)  
 (d) \_\_\_\_\_
- (e) A method cannot call itself, either directly or indirectly. (1)  
 (e) \_\_\_\_\_
- (f) `String` is a primitive type in Java. (1)  
 (f) \_\_\_\_\_
- (g) An improperly indented java file will not compile. (1)  
 (g) \_\_\_\_\_
- (h) In the expression `3 > 4 && foo(5)`, method `foo` will be called with an argument of 5. (1)  
 (h) \_\_\_\_\_
- (i) It is possible to have multiple methods with the same name in a given class. (1)  
 (i) \_\_\_\_\_
- (j) It is possible to have multiple classes with the same name in a given Java program. (1)  
 (j) \_\_\_\_\_

2. Why don't the following code snippets compile? Select the single correct answer for each.

(a) \_\_\_\_\_ (2)

```
public class MyClass {
    private double x = 3.141592;

    public static void main(String[] args) {
        System.out.println(x);
    }
}
```

- A. Cannot access private variable x from a public method.
- B. Cannot access x without an instance of MyClass.
- C. Variable x is out of scope in the main method.
- D. The println method expects a String, not an double.
- E. Some other error.
- F. This code will successfully compile.

(b) \_\_\_\_\_ (2)

```
public static boolean foo(short n) {
    int default = 10;
    return n < default;
}
```

- A. Cannot compare int and short variables.
- B. Cannot use a keyword as a variable name.
- C. Invalid parameter type.
- D. Return value does not match method return type.
- E. Did not initialize value of n.
- F. Some other error.
- G. This code will successfully compile.

(c) \_\_\_\_\_ (2)

```
int[] numbers = new int[25]
```

- A. The array is not initialized.
- B. Should be `new int(25)` on right hand side.
- C. Should use `int numbers` on left hand side.
- D. Should not use keyword `new` when declaring an array.
- E. Some other error.
- F. This code will successfully compile.

(d) \_\_\_\_\_ (2)

```
public class MyClass {
    private static final int x = 10;

    public static void main(String[] args) {
        x++;
        System.out.println(x);
    }
}
```

- A. Cannot access private variable x from a public method.
- B. Cannot access x without an instance of MyClass.
- C. Variable x is out of scope in the main method.
- D. Variable x is a constant, so cannot be incremented in main.
- E. Some other error.
- F. This code will successfully compile.

3. Write the answer in the blank provided.

(a) Name one of the integer data types in Java. (2)

(a) \_\_\_\_\_

(b) Which primitive data type could I use to store the value of the number  $\pi$ ? (2)

(b) \_\_\_\_\_

(c) What is the keyword used to make a variable or method visible to all classes? (2)

(c) \_\_\_\_\_

(d) Name one of the types of loops in Java. (2)

(d) \_\_\_\_\_

(e) What is the name for the special type of method that creates instances of a class? (2)

(e) \_\_\_\_\_

(f) What is the keyword used to call one constructor from another in the same class? (2)

(f) \_\_\_\_\_

(g) The last index of an array is 20. How many elements can the array hold? (2)

(g) \_\_\_\_\_

(h) If I declare an `int` variable but do not initialize it, what is its value? (2)

(h) \_\_\_\_\_

(i) If I declare a `String` variable but do not initialize it, what is its value? (2)

(i) \_\_\_\_\_

(j) What is the name of the method that is run first when a Java program executes? (2)

(j) \_\_\_\_\_

(k) What is the return type of the method that is run first when a Java program executes? (2)

(k) \_\_\_\_\_

(l) How many parameters does the method that is run first when a Java program executes have? (2)

(l) \_\_\_\_\_

(m) What is is called when a method calls itself? (2)

(m) \_\_\_\_\_

4. The following Java program compiles and runs. What is its output?

(14)

```
public class MethodTest {

    public static int foo(int a, String[] strs) {

        int b = a % strs.length;
        String c = strs[b];

        System.out.println("a = " + a + ", b = " + b + ", c = " + c);

        if(a < 3 * b) {
            strs[b] = "one";
        } else {
            strs[b] = "two";
        }
        return c.length();
    }

    public static void main(String[] args) {
        int a = 7;
        int b = 4;
        String[] c = new String[]{"three", "four", "five", "six"};

        System.out.println(a + ", " + foo(a, c));
        System.out.println(b + ", " + foo(b, c));

        for(int i = 0; i < c.length; i++) {
            System.out.println("at " + i + ", " + c[i]);
        }
    }
}
```

5. The following Java program compiles and runs. What is its output?

(12)

Reminder: `System.out.print` will print its argument without going to a new line afterwards.

```
public class ArrayTest {

    public static final int ROWS = 4;
    public static final int COLS = 3;

    public static void fillArray(int[][] arr) {
        int x = 0;
        for(int c = COLS - 1; c >= 0; c--) {
            for(int r = ROWS - 1; r >= 0; r--) {
                arr[r][c] = x;
                x++;
            }
        }
    }

    public static void printArray(int[][] arr) {
        for(int r = 0; r < ROWS; r++) {
            System.out.print("[ "); // bracket and space at row start
            for(int c = 0; c < COLS; c++) {
                System.out.print(arr[r][c] + " "); // value and space
            }
            System.out.println("]"); // bracket and newline at row end
        }
    }

    public static void main(String[] args) {
        int[][] grid = new int[ROWS][COLS];
        fillArray(grid);
        printArray(grid);
    }
}
```

6. The following Java program compiles and runs. What is its output?

(14)

```
public class Thing {
    private int x;

    public Thing(int x) {
        this.x = x;
    }

    public void doStuff(int x) {
        System.out.println(x);
        System.out.println(this.x);
        this.x /= x;
    }

    public static void main(String[] args) {

        int x = 6;
        Thing a = new Thing(x);
        ++x;
        a.doStuff(x+1);
        x--;
        Thing b = new Thing(x-1);
        x++;
        b.doStuff(x);
        x %= 4;
        a.doStuff(x);
        x *= 3;
        System.out.println(x);
    }
}
```

7. The following Java program compiles and runs. What are the *first* line, *third* line, and *last* line of its output? (I don't want all the output, just the lines specified.)

(6)

```
public class LoopNest {
    public static void main(String[] args) {

        String[] color = {"purple", "blue", "green",
                          "yellow", "orange", "red"};

        String[] object = {"ball", "tree", "pen",
                           "hat", "shoe", "flower"};

        int a = color.length;
        while(a > 0) {

            a--;

            for(int b = 0; b < object.length; b++) {
                String msg = color[a] + " " + object[b];
                System.out.println(msg);
            }
        }
    }
}
```

8. Write a method that takes an array of integers and returns true if each value in the array is greater than the one before and false if not. You may assume the array will be length 2 or more. (10)

I have already given you the method header so you know the parameter and return types. The following examples are to give you a feel for how it works. You must implement the method described above to work for all possible inputs, not just these examples.

```
isIncreasing(new int[] { 2, 3, 5, 9 }) → true  
isIncreasing(new int[] { 2, 3, 1 }) → false  
isIncreasing(new int[] { -5, -1, 0, 7, 9 }) → true  
isIncreasing(new int[] { 1, 1 }) → false
```

---

```
public static boolean isIncreasing(int [] nums) {
```