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	9	Total
Points:	20	7	6	12	13	12	10	10	10	100
Score:										

1. Write the answer in the blank provided.

   (a) What is the keyword used to make a variable or method visible only to the class containing it?   (2)

   (a) _____

   (b) What is the keyword used to make a variable that holds a constant value?   (2)

   (b) _____

   (c) What is the keyword used to make a variable or method belong to a *class* rather than an *instance*?   (2)

   (c) _____

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

   (d) _____

   (e) Name one of the floating point data types in Java.   (2)

   (e) _____

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

   (f) _____

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

   (g) _____

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

   (h) _____

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

   (i) _____

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

   (j) _____

2. True or false questions

   (a) An array can be returned by a method.   (1)

                                  (a) _____

   (b) An object can not be passed as a parameter.   (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) _____

3. Given the definitions below, evaluate the following boolean expressions to true or false.

```java
boolean apple = true;
boolean banana = false;
boolean kiwi = true;
boolean orange = false;
```

   (a) `!kiwi`   (1)

                                  (a) _____

   (b) `apple && banana`   (1)

                                  (b) _____

   (c) `apple || banana || orange`   (1)

                                  (c) _____

   (d) `!apple || (!banana && kiwi && orange)`   (1)

                                  (d) _____

   (e) `!apple && (!banana || kiwi || orange)`   (1)

                                  (e) _____

   (f) `(apple || orange) && (banana || orange)`   (1)

                                  (f) _____

4. Why do the following code snippets not compile? (Explain in one sentence each.)

(a)
```
int a = 1.0 / 3;
```
(3)

(b)
```
String greeting = "hello"
```
(3)

(c)
```
int[] numbers = int[25];
```
(3)

(d)
```
public class MyClass {
    private int x = 10;

    public static void main(String[] args) {
        System.out.println(x);
    }
}
```
(3)

5. The following Java program compiles and runs. What is its output?   (13)

```java
public class MethodTest {

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

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

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

        if(b == a) {
            strs[b] = "AAA";
            return "equal";
        } else {
            strs[b] = "BBB";
            return c;
        }
    }

    public static void main(String[] args) {
        int a = 2;
        int b = 7;
        String[] c = new String[]{"apple", "orange", "banana"};

        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(i + ", " + c[i]);
        }
    }

}
```

6. 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.

```java
public class ArrayTest {

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

    public static void fillArray(int[][] arr) {
        int x = 0;
        for(int c = 0; c < COLS; c++) {
            for(int r = 0; r < ROWS; r++) {
                x += 2;
                arr[r][c] = x;
            }
        }
    }

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

7. The following Java program compiles and runs. What is its output?   (10)

```java
public class Foo {
    private int x;

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

    public void printVals(int x) {
        System.out.println(this.x);
        System.out.println(x);
    }

    public static void main(String[] args) {

        int x = 2;
        Foo a = new Foo(x);
        x += 4;
        Foo b = new Foo(x);
        x--;
        --x;
        a.printVals(x);
        x -= x;
        b.printVals(x);
        x += 1;
        System.out.println(x);
    }
}
```

8. The following Java program compiles and runs. What are the *first* line, *third* line, and *last* line (10) of its output? (I don't want all the output, just the lines specified.)

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

        String[] modifier =
            new String[]{"Merry", "Survive", "Enjoy", "Happy"};

        String[] occasion =
            new String[]{"Holidays", "Christmas", "Hanukkah",
                         "Winter Break", "New Year", "Finals Week"};

        for(int a = 1; a < modifier.length; ++a) {

            int b = occasion.length - a;

            while(b >= 0) {
                String greeting = modifier[a] + " " + occasion[b] + "!";
                System.out.println(greeting);
                b--;
            }
        }
    }
}
```

9. Write a method that takes an array of strings and returns the average length of the strings in   (10)
the array, returning -1 if the array is empty.

I have already given you the method header so you know the parameter and return types. The
following examples are just to give you a feel for how it works.

averageLength(new String[]{"hello", "goodbye"}) → 6.0
averageLength(new String[]{"cat", "dog", "mouse", "bird"}) → 3.75
averageLength(new String[]{}) → -1.0
averageLength(new String[]{"", "", ""}) → 0.0
averageLength(new String[]{"I", "hope", "you", "have", "had", "fun"}) → 3.0

```java
    public static double averageLength (String [] strs) {
```