NAME: ANSWER KEY

login id:

Precept:

COS 126 Midterm 1 Written Exam Fall 2014

This test has 9 questions, weighted as indicated. The exam is closed book, except that you are allowed to use a one page cheatsheet. No calculators or other electronic devices are permitted. Give your answers and show your work in the space provided.

Print your name, login ID, and precept number on this page (now), and write out and sign the Honor Code pledge before turning in this paper. *Note*: It is a violation of the Honor Code to discuss this midterm exam question with anyone until after everyone in the class has taken the exam. You have 50 minutes to complete the test.

"I pledge my honor that I have not violated the Honor Code during this examination."

Signature

1	/8
2	/9
3	/9
4	/7
5	/9
6	/6
7	/6
8	/8
9	/8
TOTAL	/70

1. Types and Casts (8 points). Give the type and value of each of the following Java expressions, supposing that each is the argument of a `println()` call. If an expression will not compile, write `Illegal` under *type* and put an X in *value*. You must fill in every entry. Entries left blank will be marked incorrect.

expression	*type*	*value*
`(3 < 2) && (1 > 0)`	boolean	false
`"800" + 99`	String	"80099"
`800 + 99 + "A"`	String	"899A"
`3 + (int) Math.random()`	int	3
`(double) (3 / 2)`	double	1.0
`3 / 2.0 + 2 / 5`	double	1.5
`(8 <= 2) || (2e88 <= 88e2)`	boolean	false
`Integer.parseInt("8.5*2")`	Illegal	X

2. Miscellaneous (9 points). Answer the following questions.

A. Which of the following are true of standard output? Circle *all* that apply.

[✓ (i) No limit on its length.]

[✓ (ii) Can be redirected to a file.]

(iii) It is a recursive static method.

[✓ (iv) It is built in to the Java system.]

[✓ (v) It is an abstraction implemented as a library.]

(vi) Cannot use when using command-line input.

B. Which of the following best describes what is a data type? Circle the *single best* answer.

(i) A set of values.

(ii) A set of operations.

(iii) A sequence of 0s and 1s.

[✓ (iv) A set of values and operations on those values.]

(v) The arguments, name, and return value for a function.

C. Which of the following best describes dynamic programming?
Circle the *single best* answer.

(i) Susceptible to exponential waste.

(ii) A programming paradigm centered on rapid change.

(iii) The process of changing Java source code into Java bytecode.

[✓ (iv) A method for solving problems by first solving simpler subproblems.]

(v) An approach to saving memory by recomputing values when necessary.

3. **Array declarations (9 points).** Among the following code fragments, circle the ones that will *not* cause a compile-time error.

A. `int[] a = new[10];`

B. [✓ `int[] a = new int[10];`]

C. `int[10] a = new int[10];`

D. `int[] a = {1, 2, 3}; int b = a;`

E. [✓ `int[] a;`]

F. `int a = {1, 2, 3};`

G. [✓ `int[][] a = {{1, 2, 3}, {1, 2, 3}};`]

H. `int[][] a = new int[10];`

I. [✓ `int[][][] a = {{{1, 2, 3}}};`]

4. Loops and conditionals (7 points). Consider the following Java code fragment:

```
int N = a.length;
// Double.POSITIVE_INFINITY is larger than any other double
double min = Double.POSITIVE_INFINITY;
for (int i = 0; i < N; i++)
   for (int j = i+1; j < N; j++)
   {
      double delta = Math.abs(a[i] - a[j]);
      if (delta < min)
         min = delta;
   }
```

A. Suppose that the array `a[]` is initialized as follows:

```
double[] a = { 4.5, 3.5, 6.0, 20.0, 3.0 };
```

What is the value of `min` upon termination of the nested `for` loops? Circle your answer.

[✓ 0.5]   3.5 - 3.0

B. Given an array `a[]`, describe *in 15 words or less* the value of `min` after this code is executed. Assume that the length of the array is at least two.

min is the smallest absolute difference between any pair of numbers in the array a

5. Input (9 points). Consider the following Java program, which is not intended to do anything other than test your understanding of command-line and standard input:

```
public class IOfun
{
    public static void main(String[] args)
    {
        int x = 0;
        for (int i = 1; i < args.length; i++)
            x  = x + Integer.parseInt(args[i]);
        int n = Integer.parseInt(args[0]);
        String s = "";
        for (int i = 0; i < n; i++)
            s  = s + StdIn.readString() + " ";
        StdOut.println(x + s);
    }
}
```

Assume that the contents of the file `input.txt` is as shown by the following `more` command:

```
% more input.txt
now is the time
```

A. What is the output of the following command? Circle your answer.

```
% java-introcs IOfun 0 1 2 3
```

```
6
```

B. What is the output of the following command? Circle your answer.

```
% java-introcs IOfun 2 0 1 2 < input.txt
```

```
3now is
```

C. What is the output of the following command? Circle your answer.

```
% java-introcs IOfun 2 < input.txt | java-introcs IOfun 2 1 2 3
```

```
60now is
```

6. Debugging (6 points). Consider the following program, which is intended to print the words of a popular song. The song is neverending, so most of the code is enclosed in an infinite loop (imagine that it is being used to drive a teleprompter in the back of a bus).

```
 1    public class Bottles
 2    {
 3       public static main (String[] args)
 4       {
 5          string phrase =" bottles of beer ";
            int repeats = Integer.parseInt(args[0]);
 6          for (int i = 0; i < repeats; i++)
 7          {
 8             int N = 99;
 9             while ( N > 2 )
10             {
11               StdOut.println(N + phrase + "on the wall");
12               StdOut.println(N + phrase);
13               StdOut.println("Take one down, pass it around");
14               StdOut.println(N-1 + phrase + "on the wall");
15               StdOut.println();
16             }
17             // Verses for 1 and 0 (omitted)
18             StdOut.println("Go to the store, buy some more");
19             StdOut.println("99 bottles of beer on the wall");
20          }
21       }
22    }
```

In the spaces provided below, identify (*in ten words or less*, each) three bugs and describe how to fix them. In parts A. and B. fill in the blank to refer to a line in the code and circle your answers. In part C. just fill in the blanks.

A. A bug that will keep the code from compiling, on line 3 : void main

B. Another bug that will keep the code from compiling, on line 5 : String

C. A bug that will cause an infinite loop (assuming the compile-time bugs are fixed in the obvious way):

Add this code N = N -1 after this line 14 or 15

7. Methods and scope (6 points). Consider the following program, which is not intended to do anything other than test your understanding of functions, scope and side effects.

```
public class Marx
{
    public static int groucho(int i)
    {
     int j = i + 3;
     return j;
    }
    public static void chico(int[] b, int j)
    {
     int i = j + 1;
     b[groucho(i)] = b[j];
    }
    public static void main(String[] args)
    {
     int[] a = { 1, 1, 2, 3, 5, 8, 13};
     int i = 1;
     chico(a, i);
     StdOut.println("Groucho says " + groucho(i));
     StdOut.println("Harpo says " + a[groucho(i)]);
     StdOut.println("Zeppo says " + a[groucho(i+1)]);
    }
}
```

Give the output of this program in the blanks provided below.

A. First line: Groucho says 4

B. Second line: Harpo says 5

C. Third line: Zeppo says 1

8. Performance (8 points). Characterize each of the specified quantities by writing one of the words *logarithmic, linear*, *quadratic*, *cubic*, or *exponential* (with reference to a function of *N*) in each of the blanks at right below.

A. Memory use called for by the code

```
        int[][] a = new int[N][N*N];
```

cubic

B. Time required to execute the code

```
        int i = N;
        while (i > 0) i /= 2;
```

logarithmic

C. Time required to execute the code

```
        int i = N;
        while (i > 0)
        {
           int[] a = new int[i];
            i /= 2;
        }
```

linear

D. Time required to compute 2^N by calling

```
        public static int twoToThe(int N)
        {
           if (N == 0) return 1;
           return twoToThe(N-1) * twoToThe(N-1);
        }
```

exponential

E. (2 points) The order of growth of the running time of a program that runs for 30 seconds when *N* is 100,000, 1 minute when *N* is 200,000, and 1 hour when *N* is 12,000,000.

linear

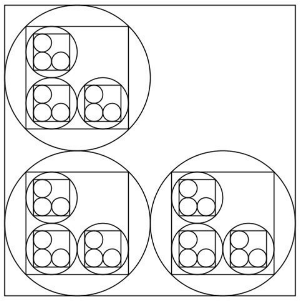
F. (2 points) Time required to execute the code

```
        String x = "hi";
        for (int i=0; i < N; i++) x += x;
```

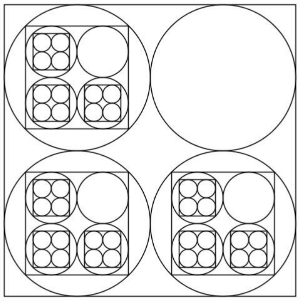
exponential

9. Recursion (8 points). Consider the following program, which includes a recursive method:

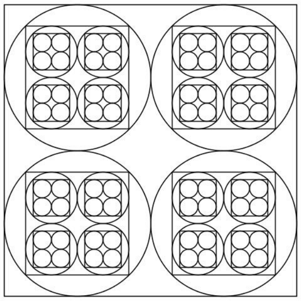
```
public class Pattern
{
    public static void draw(int n, double x, double y, double r)
    {
        if (n == 0) return;

        StdDraw.square(x, y, r);
        StdDraw.circle(x - r/2, y - r/2, r/2);
        StdDraw.circle(x - r/2, y + r/2, r/2);
        StdDraw.circle(x + r/2, y - r/2, r/2);
        StdDraw.circle(x + r/2, y + r/2, r/2);   // line 11

        double newR = Math.sqrt(r * r / 2) / 2;
        draw(n-1, x - r/2, y - r/2, newR);
        draw(n-1, x - r/2, y + r/2, newR);
        draw(n-1, x + r/2, y - r/2, newR);
        draw(n-1, x + r/2, y + r/2, newR);       // line 17
    }

    public static void main(String[] args)
    {
        draw(3, .5 , .5  , .5);
    }
}
```



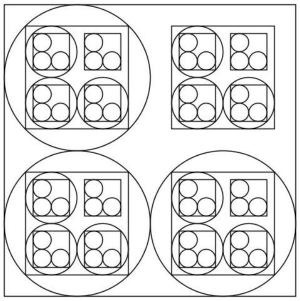
A



B



C



D

In the spaces below write A, B, C, or D to identify the drawing created by the program under the stated conditions. You must use all four letters.

No changes: C

Line 11 omitted: D

Line 17 omitted: B

Lines 11 and 17 both omitted: A