NAME: **login id:**

Precept:

COS 126 Midterm 1 Written Exam Fall 2011

This test has 8 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.

Solutions

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

____

Signature

1	/7
2	/9
3	/7
4	/6
5	/9
6	/8
7	/6
8	/8
TOTAL	/60

1. Types and Casts (7 points). Give the type and value of each of the following Java expressions. 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*
`( 7 / 2 ) * 2.0`	double	6.0
`( 7 / 2.0 ) * 2`	double	7.0
`!(!!false && !!true)`	boolean	true
`"1.1" + "2.2"`	String	"1.12.2"
`(1 < 2 < 3)`	Illegal	X
`1 + 2.2 + "1.1" + "a"`	String	"3.21.1a"
`Integer.parseInt("123")`	int	123

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

A. `int[] a = int[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}; int[] b = a;`

3. Syntax and Scope (7 points). Consider the following code.

```
public class Cubes
{
   public static int square(int i)
   {   return i * i * i;  }

   public static void main(String[] args)
   {
      for (int i = 1; i <= 1000; i++)
         StdOut.println(square(i));
   }
}
```

Among the following statements, circle those that are true.

A. Will not compile because braces in `square()` are not on separate lines.

B. Will not compile because braces are missing in the `for` loop.

C. Will not compile because `i` is not declared in `square()`.

D. Prints only a few lines because `square()` rapidly increases the `i` used by main.

E. Prints the squares of the integers from 1 to 1000.

F. Prints the cubes of the integers from 1 to 1000.

G. Goes into an infinite loop.

4. Piping (6 points). Consider the following Java program:

```
public class EZ
{
   public static void main(String[] args)
   {
      int x = StdIn.readInt();
      int y = StdIn.readInt();
      StdOut.println(y);
      StdOut.println(x % y);
      StdOut.println(x);
   }
}
```

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

```
% more input.txt
83 14
```

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

```
% java EZ < input.txt
```

```
14

13

83
```

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

```
% java EZ < input.txt | java EZ
```

```
13

 1

14
```

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

```
% java EZ < input.txt | java EZ | java EZ
```

1

0

13

5. Debugging (9 points). Consider the following code snippet, which is intended to examine an array of `double` values and print out the minimum value in the array and its location. The line numbers at left are for your convenience in referring to the code. You may assume `arr[]` is an array of `double` values that has been properly declared and initialized.

```
 1      int N = arr.length;
 2      double min = 0;
 3      int minLocation=0;

 4      for(int i = 1; i <= N; i++)
 5      {
 6         if( arr[i] < min )
 7            min = arr[i];
 8            minLocation = i;
 9      }

10      System.out.print("The minimal value is arr[");
11      System.out.println(minLocation + "] = " + min);
```

In the spaces provided below, identify (*in ten words or less,* each) three bugs in the code that will result in either an incorrect answer being printed or a program crash. Circle your answers.

A. A bug:

line 2: min = 0 fails if all items in arr[] > zero.

B. Another bug:

Curly braces needed around lines 7-8.

C. A third bug:

Line 4: i <= N results in ArrayIndexOutOfBounds crash.

6. Methods (8 points). For the purpose of this question, assume that you are developing static methods for your own statistics library `MyStats`.

A. (4 points) Write a public static method `threeEqual()` for `MyStats` that takes three `int` values as arguments and returns `true` if all three numbers are equal, `false` otherwise.

```
public static Boolean threeEqual(int a, int b, int c) {
    return (a==b) && (b==c);
}
```

B. (2 points) Give the *signature* of a public static method `median()` for `MyStats` that is to take as an argument an array of `int` values and return the middle value. Do not give the code, just the signature (a single line of code).

```
public static int median(int[] values)
```

C. (2 points) Give a single Java expression that a client of `MyStats` could use to test whether the medians of three arrays of `int` values `int[] a`, `int[] b`, and `int[] c` are all equal.

```
MyStats.threeEqual(MyStats.median(a), MyStats.median(b), MyStats.median(c))
```

7. Performance (6 points). A Python programmer experiences the following approximate running times for a program that reads N numbers, for various values of N.

N	*time*
1000	just over 1 second
10000	just under 2 minutes
100000	about 3 hours
1 million	?

A. Which of the following best describes the likely running time of this program if our programmer were to run it for N = 1 million? Circle your answer.

a. About a day

b. About two weeks

c. About three months

d. About four years

B. Which of the following best describes the order of growth of the running time of this program? Circle your answer.

a. Logarithmic

b. Linear

c. Linearithmic

d. Quadratic

e. Cubic

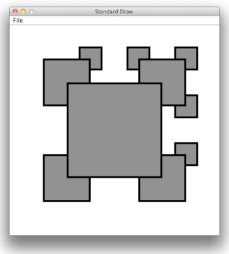
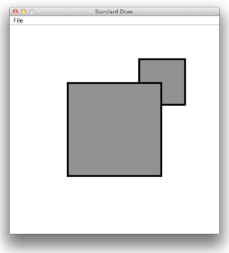
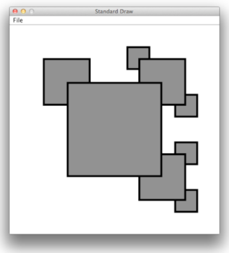
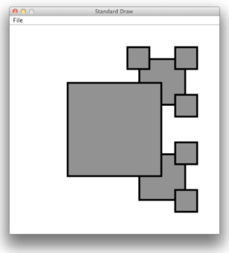
Every time N increases by a factor of 10, time increases by a factor of about 100.

8. Recursive Graphics (8 points). Consider the following recursive method:

```
public static void squareLife(double x, double y, double r)
{
   if (r < 0.05) return;

   //Select a random integer from the set [0, 1, 2, 3, 4]
   int randomInt = (int) (Math.random() * 5);

   if (randomInt < 1)
      squareLife(x-r, y-r, r/2);    //bottom left
   if (randomInt < 2)
      squareLife(x-r, y+r, r/2);    //top left
   if (randomInt < 3)
      squareLife(x+r, y-r, r/2);    //bottom right
   if (randomInt < 4)
      squareLife(x+r, y+r, r/2);    //top right

   drawShadedSquare(x, y, r);
}
```

Assume that the helper method method `drawShadedSquare()` draws a gray shaded square of radius `r` that is outlined in black and centered on `(x, y)`.

A. Suppose that a client program issues the call `squareLife(0.5, 0.5, 0.25).` Which of the following figures could possibly result? Answer by circling 1, 2, or 3 of the figures.

this one   this one

B. Suppose that a client program issues the call `squareLife(0.5, 0.5, 0.6).` What is the maximum possible number of times `drawShadedSquare()` could be called as a result? Write your answer in the blank provided.

Maximum number of squares drawn: ________85__________