Name: _____

CS-231 Exam 2 Spring 99

Score:	1.	(max 12)	5.	(max 9)
	2.	(max 12)	6.	(max 20)
	3.	(max 10)	7.	(max 18)
	4.	(max 9)	8.	
	Total:		(max 90)	

1. (6 pts each, 12 pts total) Using Java syntax, write a boolean expression that evaluates to true when

   (a) The number x is *outside* the range from 15 to 50.

   (b) W is not equal to X or Y

2. (6 pts each, 12 pts total) Assume A and B are boolean variables. Simplify the following boolean expression:

   (a) `A && !A`

   (b) `!( !A || B ) && A`

3. (10 pts) Recursion: Write a *recursive* method `factorial(n)` to compute the factorial of a number `n`. You should not have any loops in your code. For example, to print the factorial of 10, we should be able to write call your recursive method as follows:

```
System.out.println("The factorial of 10 is " + factorial(10));
```

4. (9 pts) What is the output of the following code:

```
int i, j, t, k = 0;
for (i = 0; i < 10; i++)
{
   for (j = 0; j < 20; j++)
   {
       for (t = 0; t < 30; t++)
         k++;
   }
}
System.out.println("i = " + i +
                   " j = " + j +
                   " t = " + t +
                   " k = " + k );
```

5. (9 pts) What is the output of the code given below for the following values of `day` (be careful!):

   (a) day = 0

   (b) day = 2

   (c) day = 8

```
double rate = 0;
switch (day)
{
    case 0:
       System.out.println("Case 0");
       rate = 10;
       break;
    case 1:
       System.out.println("Case 1");
       rate = 20;
    case 2:
       System.out.println("Case 2");
    case 3:
       System.out.println("Case 3");
       rate = 40;
    case 4:
       System.out.println("Case 4");
       rate = 50;
       break;
    case 5:
       System.out.println("Case 5");
       rate = 60;
    default:
       System.out.println("default");
}
System.out.println("The rate is " + rate);
```

6. (20 pts total) Write for-loops to do the following.

   (a) (10 pts) Sum the odd numbers in the range from 1 to 100.

   (b) (10 pts) First, declare a 2D integer array called `table` with 4 rows and 5 columns. Then, use for-loops to initialize *every* element of the array `table` to the product of it's row and column. For example, `table[3][2]` should be initialized to 6 since 6=3*2.

7. (6 pts each, 18 pts total) Draw a picture of the memory after the following instructions. Include any references (arrows) even if they don't point to anything.

   (a) `int [] hours = { 10,3,4,5};`

   (b) `int temperatures[][] = new int[3][];`

   (c)
```
Complex c[][] = new Complex[3][2];
for (int i = 0; i < 2; i++)
    c[1][i] = new Complex();
```