

# CS61BL Quiz 01

Name:

SID:

Section Number:

☐ 01 ☐ 02 ☐ 03 ☐ 04 ☐ 05 ☐ 06 ☐ 07 ☐ 08 ☐ 09 ☐ 10 ☐ 11 ☐ 12

Write your name and Cal ID number above. Bubble in your section number. Complete this quiz without using a Java interpreter, and without consulting anyone other than CS 61BL staff.

## 1. Buggy Leap Year

The following **buggy** code segment was proposed as a solution to the leap year exercise from lab 1. (Recall that a leap year is either divisible by 400, or divisible by 4 but not by 100.)

```
if (year % 400 != 0) {  
    System.out.println (year + " is not a leap year");  
} else if (year % 100 != 0) {  
    System.out.println (year + " is a leap year");  
} else if (year % 4 != 0) {  
    System.out.println (year + " is not a leap year");  
} else {  
    System.out.println (year + " is a leap year");  
}
```

Put an X in the correct boxes in the table below to reveal the mistakes this program makes. The first two columns have been filled out for you.

	This is a leap year	This is not a leap year	The program says this is a leap year	The program says this is not a leap year
1900		X		
1904	X			
1905		X		
2000	X			

## 2. Do you Git it?

Recall the Git commands `init`, `add`, `commit`, `status`, `log`, `show`, `clone`, `pull`, and `push`. Bubble in the circle next to the command which allows you to...

1. tell Git to start tracking a file:

☐ `init`   ☐ `add`   ☐ `commit`   ☐ `status`   ☐ `log`   ☐ `show`   ☐ `clone`   ☐ `pull`   ☐ `push`

2. save a snapshot of the files being tracked:

☐ `init`   ☐ `add`   ☐ `commit`   ☐ `status`   ☐ `log`   ☐ `show`   ☐ `clone`   ☐ `pull`   ☐ `push`

3. see what files have changed since your last commit:

☐ `init`   ☐ `add`   ☐ `commit`   ☐ `status`   ☐ `log`   ☐ `show`   ☐ `clone`   ☐ `pull`   ☐ `push`

4. see a list of previous commits:

☐ `init`   ☐ `add`   ☐ `commit`   ☐ `status`   ☐ `log`   ☐ `show`   ☐ `clone`   ☐ `pull`   ☐ `push`

5. create a local copy of a remote repository:

☐ `init`   ☐ `add`   ☐ `commit`   ☐ `status`   ☐ `log`   ☐ `show`   ☐ `clone`   ☐ `pull`   ☐ `push`

6. send your locally tracked files to a remote repository:

☐ `init`   ☐ `add`   ☐ `commit`   ☐ `status`   ☐ `log`   ☐ `show`   ☐ `clone`   ☐ `pull`   ☐ `push`

### 3. MultiSwap

```
public void swap(int x, int y) {
    int temp = y;
    y = x;
    x = y;
}

public static void multiplyBy2(int[] A) {
    int[] B = A;
    for (int i = 0; i < B.length; i++) {
        B[i] = B[i] * 2;
    }
}

public static void multiplyBy5(int[] A) {
    int[] temp = new int[A.length];

    for (int i = 0; i < temp.length; i++) {
        temp[i] = 5 * A[i];
    }

    A = temp;
}

public static void main(String[] args) {
    int a = 5;
    int b = 10;
    swap(a, b);

    int[] arr;
    arr = new int[]{1, 2, 3, 4};
    multiplyBy2(arr);

    arr = new int[]{1, 2, 3, 4};
    multiplyBy5(arr);
}
```

What are the values of **a** and **b** after main is run?

**a** : \_\_\_\_\_      **b** : \_\_\_\_\_

What is the value of **arr** after *multiplyBy2* is run?

**arr** : { \_\_\_\_\_ }

What is the value of **arr** after *multiplyBy5* is run?

**arr** : { \_\_\_\_\_ }