

You only need to write your name and e-mail ID on the first page.

This exam is CLOSED text book, closed-notes, closed-calculator, closed-neighbor, etc. Questions are worth different amounts, so be sure to look over all the questions and plan your time accordingly. Please sign the honor pledge here:

---



---



---



---

Page 1	___ / 2
Page 2	___ / 15
Page 3	___ / 10
Page 4	___ / 8
Page 5	___ / 18
Page 6	___ / 16
Page 7	___ / 9
Page 8	___ / 22
<b>Total</b>	<b>___ / 100</b>

*Note: When an integer type is required use `int`, when a floating-point type is required use `double`. If we don't specify an aspect of the problem, you can choose it.*

**Note: If you are still writing on the exam after “pens down” is called – even if it is just to write your name – then you will receive a zero on this exam. No exceptions!**

1. [2 points] What lab section are you in?

\_\_\_ CS 101-E

\_\_\_ CS 101-4 (lab 2:00–3:30 p.m. Fri)

\_\_\_ CS 101-2 (lab 7:00–8:30 p.m. Thu)

\_\_\_ CS 101-5 (lab 10:00–11:30 a.m. Fri)

\_\_\_ CS 101-3 (lab noon–1:30 p.m. Fri)

\_\_\_ CS 101-6 (lab 2:00–3:30 p.m. Thu)

2. [3 points] Write a single Java statement that declares a variable called *count* to hold a value of type *int*. That variable should be initialized to 2006 at the end of the statement.

3. [3 points] Write the Java code needed to declare a *public, static* method called *compute()* that takes no parameters and returns a value of type *int*. Write your answer in the blank area. (The next question asks you to fill in code between the braces. Leave it blank for now.)

```
_____ {  
    // this part filled in in the next question  
}
```

4. [3 points] Complete the Java code for the last question to provide an implementation of the method, *compute()*, that always returns 0 (zero). Write your answer below:

5. [3 points] Write the Java code to assign the value returned by the method *compute()* to the variable *count*. We'll assume that the *compute()* method is in a class called *Exam3*, and that the code you are to write below is in a *different* class.

6. [3 points] Suppose that the *int* variables *a*, *b* and *c* have the values 1, 2 and 3, respectively and that the following Java statements are then executed. What are the values of *a*, *b* and *c* after these statements are executed?

```
a = a + 1;  
b = a + b + 1;  
c = a + b + c + 1;
```

7. [3 points] Suppose that the integer variable *max* holds a positive integer value (such as 3, for example). Complete the following code by filling in the blank with a `for` loop so that the code, when executed, prints out integer values starting with 0 (zero) and ending with the value of *max*. (If the value of *max* is 3, for example, then the code should print out "0 1 2 3".) Assume that the appropriate Java libraries have already been imported.

```

_____ {
    System.out.print(i + " ");
}

```

8. [3 points] Complete the following code by filling in the blank with a `while` loop so that, when executed, the program prompts the user to enter an integer value and keeps on prompting the user to enter integer values until the user enters a 0 (zero), at which time the `while` loop should terminate. Assume that all the appropriate Java libraries have been imported and that the variable *stdin* holds a value of type *Scanner* for acquiring the user's inputs.

```

boolean done = false;

_____ {
    System.out.print("Enter an integer value: ");
    int answer = stdin.nextInt();
    if (answer == 0)
        done = true;
}

```

9. [4 points] You can multiply two positive integers by starting with 0 (zero) and adding 1 (one) the correct number of times. For example,  $3 * 4 = (1 + 1 + 1) + (1 + 1 + 1) + (1 + 1 + 1) + (1 + 1 + 1)$ , which is just three four times, of course. Complete the following code by filling in the blank lines with *nested for* loops so that after the code runs the value of the variable *product* is  $m1 * m2$  (i.e. *m1* times *m2*).

```

// at this point, m1 and m2 store positive integer values
int product = 0;

_____ {

    _____ {
        product++;
    }
}

// at this point, product stores the value m1*m2

```

10. [4 points] Suppose that you're writing a class, *MyMath*, that will provide methods for use by other classes for doing simple mathematical calculations. One method, *multiply()*, takes two positive integer values and returns an integer (namely, the product). You could, for example, use the code that you wrote for the last question to implement this method. Fill in the blank in the following code to declare the *multiply()* method. You do not need to provide an implementation. Note that this method should be `public` and `static`.

```
class MyMath {  
  
    _____ {  
  
        // some implementation would go here, but is not required for  
        // this question  
    }  
}
```

11. [4 points] Suppose that you're the chief programmer for the *MyMath* class. Your customers are complaining that some of the methods are too slow. Unfortunately, you have only a little time to make improvements. Your strategy is to improve the implementations of only the most used methods, but you don't know which methods they are. What you need to do is to count the number of times that each method in the class is called. Complete the code below by declaring an integer data member, *multiplyCount*, that will keep track of the number of times the *multiply()* method is called.

```
class MyMath {  
  
    _____ // declare count here  
  
    <your multiply method declaration from the question 10 goes here> {  
        // some implementation would go here  
        count++; // increment the number of times method called  
    }  
}
```

12. [8 points] What value and type of value are produced by evaluating each of the following Java expressions? For example: for the expression  $1 + 2$ , the correct answer is  $3$ , *int*

- |                |               |                                   |               |
|----------------|---------------|-----------------------------------|---------------|
| a. $1 + 3$     | _____ , _____ | h. $1 == 0$                       | _____ , _____ |
| b. $1 + 3.0$   | _____ , _____ | i. $true \ \&\& \ false$          | _____ , _____ |
| c. $2 + 3 * 4$ | _____ , _____ | j. $true \    \ false$            | _____ , _____ |
| d. $3 / 4$     | _____ , _____ | k. $true \ ? \ true \ : \ false$  | _____ , _____ |
| e. $5 / 4$     | _____ , _____ | l. $false \ ? \ false \ : \ true$ | _____ , _____ |
| f. $5.0 / 4$   | _____ , _____ | m. $24 \% 5$                      | _____ , _____ |
| g. $1 == 1$    | _____ , _____ | n. $24 \% 5 / 2$                  | _____ , _____ |

13. [10 points] In many programming situations, you have to produce an outcome that depends on a number of conditions. Suppose, for example, that you have two conditions, represented by the `boolean` variables *pressureHigh* and *valveStuck*. Based on these conditions, you have to set the value of an integer variable, *action*, as specified by the following table. You can assume the actions are represented by integer constants (e.g., NORMAL is 0 or some other integer value).

<b>pressureHigh</b>	<b>valveStuck</b>	<b>action</b>
false	false	NORMAL
false	true	REPAIR
true	false	RELEASE
true	true	EVACUATE

Write Java code using *nested if* statements to set the value of *action* based on the values of *pressureHigh* and *valveStuck*. The code should assign these values in the same order that they appear in the table (to ease the task of verifying that the code is correct).

14. [4 points] For each of the following, indicate whether it is a *class (C)*, a *static data member (SDM)*, a *non-static data member (NDM)*, a *static method (SM)*, or a *non-static method (NSM)*. Assume that we're using the standard Java libraries.

- |                                          |                                        |
|------------------------------------------|----------------------------------------|
| a. <code>System.out.println</code> _____ | d. <code>Integer.toString</code> _____ |
| b. <code>System.out</code> _____         | e. <code>Color.white</code> _____      |
| c. <code>System</code> _____             | f. <code>Math.PI</code> _____          |

15. [4 points] Write a single Java statement to declare a variable *iarray* to be an array of 5 integers. We don't care what the initial values of the array elements are for this question.

16. [4 points] Write a `for` loop to initialize the elements of this array (*iarray*) so that the value of the first element is 1 (not 0), the second element 2, the third element 3, and so forth, through the last element. Do not use a literal 5 in your `for` loop. Instead, get the length of the array using the *length* property of the array. Be careful with your loop indices and array values!

17. [4 points] What output does the following program produce?

```
class Exam3 {
    public static void main(String args[]) {
        int iarray[] = { 1, 2, 3 };
        foo(iarray[2]);
        System.out.println(iarray[2]);
        bar(iarray);
        System.out.println(iarray[2]);
    }
    public static void foo(int i) {
        i = 5;
    }
    public static void bar(int i[]) {
        i[2] = 5;
    }
}
```

**Answer:**

18. [17 points total] This problem has several parts. The overall goal is to produce a class called *Rational* for representing rational numbers (fractions with integer numerators and denominators). Here is an outline of the class. The comments indicate the tasks you are to perform. Give your answers under the appropriate sub-questions, below.

```
public class Rational {
    // declare and initialize data members: integer numerator and denominator
    // write a default constructor initializing value to 1/1 (one over one,
    //     i.e., 1)
    // write a constructor taking non-negative integers n, d, initializing
    //     value to n/d
    // write a method toString() that returns a string to representing the
    //     given value
    // write a method mult() taking a Rational and returning the product of
    //     the two
}
```

- a. [3 points] Write the data member declarations (i.e. the instance variables). To make this test question easier, you can have them be public so that you don't have to use accessors and mutators.
- b. [3 points] Write the default constructor. It should initialize the value to 1/1.
- c. [3 points] Write the constructor taking integers  $n$  and  $d$  (which we require and assume to be non-negative) and initializing the value to  $n/d$ .

- d. [4 points] Write the method *toString()*. Given a *Rational* with value  $n/d$ , the method should return the string “ $\langle n \rangle / \langle d \rangle$ ”, where  $\langle n \rangle$  is the value of  $n$  and  $\langle d \rangle$  is the value of  $d$ . So, for example, if the value of an integer is  $3/4$ , this method should return the String “ $3/4$ ”.
- e. [4 points] Write the method *mult()* taking a *Rational* object  $r$  as a parameter and returning a *Rational* whose value is the product of  $r$  and *this*. So, for example, if you have a *Rational*  $r1$  with value  $1/2$  and a *Rational*  $r2$  with value  $1/4$ , then the expression  $r1.mult(r2)$  must return a *Rational* whose value is  $1/8$ . Don't worry about reducing fractions for this question (i.e.  $2/3 * 1/2$  can yield  $2/6$ ).
19. [7 points] What is the difference between a static variable and a non-static variable? Explain in about 25 words or less.
20. [7 points] What is the difference between public visibility and private visibility? Explain in about 25 words or less.