

Exam I

Given: 28 September 2017

Due: End of session

This exam is closed-book, closed-notes, no electronic devices allowed. The exception is the “sage page” on which you may have notes to consult during the exam. Answer questions on the pages of the exam. Do not unstaple the pages of this exam, nor should you attach any other pages to the exam. You are welcome to use the blank space of the exam for any scratch work.

Your work must be legible. Work that is difficult to read will receive no credit. Do not dwell over punctuation or exact syntax in code; however, be sure to indent your code to show its structure.

You must sign the pledge below for your exam to count. Any cheating will cause the students involved to receive an F for this course. Other action may be taken. If you need to leave the room for any reason prior to turning in your exam, you must give your exam and any electronic devices with a proctor.

You must fill in your identifying information correctly. Failure to do so is grounds for a zero on this exam. When you reach this point in the instructions, please give the instructor or one of the proctors a meaningful glance.

Print clearly the following information:	
Name (print clearly):	
Student 6-digit ID (one digit per box):	<input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/> <input type="text"/>
Your answers below tell us where to return your graded exam.	
What time do you arrive in studio/lab? (circle one) 11:30 1:00 2:30 4:00	
Which Urbauer lab? (your best guess, circle one) 214 216 218 222	

Pledge: On my honor, I have neither given nor received any unauthorized aid on this exam.

Signed: _____
(Be sure you filled in your information in the box above!)

Problem Number	Possible Points	Received Points	Grader Initials
1	20		
2	30		
3	20		
4	30		
Total	100		

1. (20 points)

(a) (10 points) Circle the correct type for each expression in the table below, and state the result of evaluating the expression:

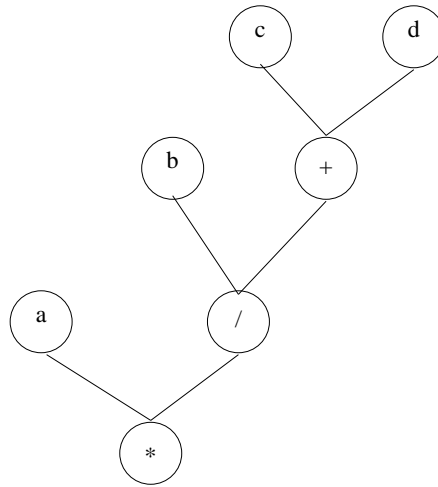
Expression	Type	Result
$7 / 2$	double int boolean String	_____
$7 / 2.0$	double int boolean String	_____
$131 > 132$	double int boolean String	_____
$1.8 + "0"$	double int boolean String	_____
<code>"Sept " + (20 + 3/2)</code>	double int boolean String	_____
$(17/8) + "" + (3 < 4)$	double int boolean String	_____
<code>true && false</code>	double int boolean String	_____
<code>! ((1 > 2) && true)</code>	double int boolean String	_____
$3/2 \leq 1$	double int boolean String	_____
<code>true false</code>	double int boolean String	_____

(b) (5 points) Below draw the expression tree¹ for the expression

$$1 + 2 * 3 + 4 * 5$$

¹Or, explain exactly the order in which the operations occur.

(c) (5 points)



Complete the blanks below regarding the tree shown above, which uses the arithmetic operators $+$, $*$, and $/$:

- The _____ operator is the first operation to execute.
- The _____ operator is the last operation to execute.

2. (30 points)

(a) (10 points) Complete the code below so that it computes and prints the integer division of $a/(a-b)$, if $a-b$ is not zero. If $a-b$ is zero, the code should print **Error**.

```
int a = ap.nextInt("Value for a?");  
int b = ap.nextInt("Value for b?");
```

Continued on next page...

- (b) (20 points) Complete the code below so that it determines how many of N random numbers are less than 0.5, with each random number chosen by a call to `Math.random()`. Recall that each call to `Math.random()` returns a double `r` such that $0 \leq r < 1$. Do not use any arrays! Your code should produce output such as the following (in the example below, I typed 100 in response to the prompt):

```
You asked for 100 random numbers.  
Of those, 63 were less than 0.5.
```

Your output will depend on the value of N supplied when the program is run, as well as the results of the calls to `Math.random()`.

```
int N = ap.nextInt("How many random numbers?");
```

3. (20 points) We have studied 4 basic data types in the first part of our course: `int`, `double`, `boolean`, and `String`. Fill in the table below to supply the data type most appropriate for the specified scenario and provide one simple example. Note that a given scenario may have more than one correct answer.

Scenario	Circle one type 2 points	Example 3 points
The name of the TA most helpful to you in this course	double int boolean String	
The number of studios you have completed so far	double int boolean String	
The fraction of students who have turned in Lab 1 on time	double int boolean String	
Whether or not you have pulled from <code>git</code>	double int boolean String	

4. (30 points) An integer array `nums` has been created for your use in this problem. Each integer in the array could be positive, negative, or zero. An example of such an array follows:

<code>nums</code> array		[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
contents		50	81	5	-2	-2	-1	0	2

While the array in the example above has 8 entries, it is only an example for illustrative purposes. The array `nums` could be of any size.

- (a) (5 points) Complete the code below so that it sets `N` to the size of the `nums` array:

```
int N = nums.length;           // Our free gift to you
                                // Please, no need to thank us!
```

- (b) (10 points) Complete the code below so that it computes the sum of all of the entries in the `nums` array. Given the example above, your code would produce
The sum is 133.

```
//
// you can use N, as computed above,
// or not, as you wish
//
for (                               ) {

}
System.out.println("The sum is " + sum);
```

Continued on next page...

- (c) (10 points) Your task now is to create a `sums` array that contains at each element i the sum of all entries in `nums` up to and including entry i .

For example, given the `nums` array reprised from above:

<code>nums</code> array		[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
contents		50	81	5	-2	-2	-1	0	2

the code you are about to write would produce the `sums` array:

<code>sums</code> array		[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]
contents		50	131	136	134	132	131	131	133

Below, write the code to compute the `sums` array as described above, assuming you are given a `nums` array of some size (not necessarily the size of our running example). Do not print out anything, just compute the `sums` array properly. Be sure to declare and allocate the `sums` array.

Continued on next page...

- (d) (5 points) Finally, given the `sums` array as described above, write code below that prints each index i of `sums` at which the entry 131 appears.² In our running example, the output would be:

At 1

At 5

At 6

²If you had trouble with the previous part of this problem, just assume that `sums` is somehow computed correctly and you can still answer this part.

