First name (color-in initial)	A	B	C	D	E	F	**G**	H	I	J	K	L	**M**	N	O	P	Q	R	**S**	T	U	V	W	X	Y	Z	section (4, 5 or 6)	first name initial	last name initial
Last name (color-in initial)	A	B	C	D	E	F	**G**	H	I	J	K	L	**M**	N	O	P	Q	R	**S**	T	U	V	W	X	Y	Z			

H01: Due Thursday, 01.08 in Lab

Variables, Types (double vs. float, primitive vs. reference etc.) Instance Variables, Methods (HFJ Ch3,4)
Assigned: Mon 01.05 Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE,
or offered in person, for in person grading, during instructor or TAs office hours.
See the course syllabus at https://foo.cs.ucsb.edu/56wiki/index.php/W15:Syllabus for more details.

---

(1) (10 pts) Fill in the information below. Also, fill in the A-Z header by

- **coloring in** the first letter of your first and last name (as it appears in Gauchospace),
- writing **either 4, 5, or 6** to indicate your **discussion section (lab)** meeting time
- writing your **first and last initial** in large capital letters.

All of this helps us to manage the avalanche of paper that results from the daily homework.

name:	
umail address:	@umail.ucsb.edu

If you collaborated with AT MOST one other person on this homework, write his/her name below. She/he should also have your name on his/her paper.

Reading Assignment: '

Throughout the quarter, when I refer to **HFJ**, this means your Head First Java, 2nd Edition textbook.

- Read HFJ, Chapter 3 (especially pages 59-62) and reading notes at HFJ:Chapter_3
- Read Chapter 4 and reading notes at HFJ:Chapter_4

(2) Based on your reading in HFJ Chapter 3:

- (4 pts) If I write 3.4, is that of type double, or float?
- (4 pts) Declare x as a double and assign it the value 3.4 (as a double)
- (4 pts) Declare y as a float and assign it the value 3.4 (as a float)

(3) Variables that represent a primitive type (e.g. `boolean x;` or `int y;`) and variables containing object references (`String w;` or `Student z;`) have this in common—they are both composed of bits in memory.

But—as explained in HFJ Chapter 3—they differ in what the bits *actually* represent. You won't get this one by just guessing—you really have to read the book.

- (4 pts) What do the bits that represent `int y;` represent?
  Assume that y is assigned the value 13

- (4 pts) What do the bits that represent `String w;` represent?
  Assume that w is assigned the value "foo".

(4) Consider these questions about memory—answers are in Chapter 3 of HFJ.

- (2 pts) Does the amount of memory taken up by an object reference differ for different kinds of objects (say String vs. ArrayList<String>?)

- (2 pts) Does the amount of memory taken up by the object itself differ for different kinds of objects (assuming the same JVM)

- (2 pts) Can the amount of memory taking up for an object reference for a object particular type (say `String`) differ from one JVM to another?

(5) Based on your reading in HFJ Chapter 3, p. 59-62 and HFJ Chapter 4 p. 84:

- (4 pts) Suppose I have a class called Student. How do I declare and allocate space for a plain old Java array called `students` that can hold 5 references to Student objects?

- (5 pts) Java `for` loops look pretty much just like C++ `for` loops (see HFJ page **10** if you really need to check. Given that, assuming there is a default constructor `Student()` that you can call to create a new `Student` object, write a `for` loop that initializes all of the elements of the array `students` (from the previous problem) to be instances of the `Student` class.

- (5 pts) In C++, the name of a plain old array of `Student` objects is not an object, but is rather a pointer to a `Student` (i.e. it is of type `Student *`. What about in Java—is an array an object, yes or no?