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 (10,or 11)	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			

H03: Due Tuesday 04.09 in Lecture. Total Points: 50

Variables, and JAR files (HFJ Ch3, Ch17 (only pp 581-595)

MAY ONLY BE TURNED IN DURING Lecture ON Tuesday 04.09, 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/S13: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 would appears in Gauchospace),
- writing **either 10,11** to indicate your **discussion section** 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

Reading Assignment:

Throughout the quarter, when I refer to **HFJ**, this means your Head First Java, 2nd Edition textbook. The other textbook for the course is the Java Pocket Guide, which I'll refer to as **JPG**. Each of these has its own page on the wiki with reading notes.

- Read HFJ, Chapter 3 (especially pages 59-62) and reading notes at HFJ:Chapter_3
- Read HFJ, Chapter 17 , ONLY pages 581 through 595, and reading notes for those pages at HFJ:Chapter_17

Why are we skipping ahead?—I know that students sometimes get nervous when we "skip ahead" in the textbook, so I try to do it only when there's a really good reason. This week, there is.

The material in Chapter 17 is all about "Releasing your Code"—i.e. making it available to the rest of the world. Pages 581-595 describe how to put code into packages, and JAR files. This material doesn't really depend on any of the material in Chapters 5-16 (at least not very much.) And since the emphasis in this course is on producing "open source" software, I want to be sure that we are comfortable with making JAR files and putting our code in packages. Chapter 17 should help make some of what we are doing in lab with those topics a bit more clear.

The rest of the chapter is about Java Web Start—we'll either cover that later when we cover GUIs, or just skip it altogether.

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

- (3 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?
- (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) (5 pts) Based on your reading in HFJ Chapter 17:

To be able to run a jar file directly with `java -jar blah.jar` it is necessary for the jar file to contain a "manifest". What **is** a manifest, and **what information** does it contain?

(6) (5 pts) From Chapter 17: Briefly, what is the purpose of having multiple "packages" in Java?

(7) (5 pts) From Chapter 17: How do you indicate in the source code that a class is part of a package?