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 (9,10,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			

H11: Due Wednesday 05.02.2012 in Lecture. Total Points: 50

MAY ONLY BE TURNED IN DURING Lecture ON Wednesday 05.02.2012, 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/S12: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 9,10,11** 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:

- HFJ:Chapter_14, starting on p. **429** Serialization and File-IO: Saving Objects
- In addition to HFJ (Head First Java). The other textbook for the course is the Java Pocket Guide, which I'll refer to as **JPG**. Both HFJ and JPG have their own pages on the wiki with reading notes.
- In JPG, read Chapter 11, which is about the same topics as Chapter 14 in HFJ, i.e. Input and Output (especially File IO and Serialization).
  - You may skip the Section titled "Socket Reading and Writing" on pages 101-103. But read all of the rest.

Based on your reading in HFJ Chapter 14 and JPG Chapter 11:

(2) (4 pts) How do you indicate that part of an object that implements the Serializable interface should NOT be saved?

(3) (10 pts) Write a brief main method that writes the string "Hello Disk!" to a file called "myFirstFile.txt".

(4) Figure 11-1 on p. 98 of JPG shows an inheritance chart with four base classes: Reader, Writer, InputStream and OutputStream, and various kinds of derived classes that inherit from these.

(a) (6 pts) Fill in the blanks: According to the text, Reader and Writer (and implicitly, the classes that inherit from them)

are used mainly for _____, while

InputStream and OutputStream are used mainly for _____

(b) (5 pts) According to JPG, what is the situation where a Buffered version of an InputStream or OutputStream is appropriate?

(5) (5 pts) The File class in Java provides methods that correspond to several Unix commands. What is the equivalent java code to the Unix command:

```
mv foo.txt bar.txt
```

Note: You may need more than just the information in the chapters—you might also need to consult the Javadoc for the File class online, and maybe even try out your code to see if it works. I dont need the whole class—just write the code that would appear inside a main() method:

(6) (10 pts) For this question, consider the reading in HFJ about serialization.

Your friend B. C. Dull says:

"I don't get why object serialization is such a big deal. You have a pointer to each object, and you know how many bytes it takes up in memory, and you know the objects type. Just write those bytes to a file, along with a few extra bytes indicating the type—problem solved. Then you read those bits back in and restore the objects. What's the big deal". You however, see more deeply into the situation, and say:

"Well, B.C., it isn't quite that simple. You are forgetting a few subtle issues—things you should realize from your previous study of how data structures work—things you should have learned in CS16 and CS24. Even though those were C/C++ courses, the same problems are going to arise in Java." B.C. says: "I don't see what you are getting at. Can you explain it to me?"

What do you say to B.C. to help him/her realize why Object Serialization is more subtle than just writing out all the bits exactly as they are, and reading them back in exactly as they are?

HINT: Think about things like linked lists, pointers, and references, and how they are implemented in both C++ and Java.