

<b>Name:</b> <i>(as it would appear on official course roster)</i>	
<b>Umail address:</b>	@umail.ucsb.edu
Optional: name you wish to be called if different from name above.	section 4, 5, or 6
Optional: name of "homework buddy" (leaving this blank signifies "I worked alone")	

You may collaborate on this homework with AT MOST one person, an optional "homework buddy".

## H05: Due Wednesday, 01.13 in Lecture

**Constructors, and Primitive Variables vs. Object References on the Stack and Heap (HFJ Ch9)**

Assigned: Wed 01.06      Total Points: 50

MAY ONLY BE TURNED IN IN THE LECTURE/LAB LISTED ABOVE AS THE DUE DATE,  
OR IF APPLICABLE, SUBMITTED ON GRADESCOPE. There is NO MAKEUP for missed assignments;  
in place of that, we drop the five lowest scores (if you have zeros, those are the five lowest scores.)

# 1

# H05

CS56 W16

### Reading Assignment:

In HFJ, Review HFJ:Chapter\_7 and HFJ:Chapter\_8 then read HFJ:Chapter\_9, which describes a **major difference** between C++ and Java: the issue of **garbage collection**. This is a *crucial* chapter, so read it *carefully*. If there are reading notes on the wiki, consult those too—sometimes they contain helpful hints.

- (6 pts) Fill in the homework header properly—this helps us keep the grading pipeline flowing so that you get credit for your work and get feedback more quickly.
  - writing **either 4, 5, or 6** to indicate your **discussion section (lab)** meeting time
  - entering **BOTH** your name AND your umail address EVERY time.

**Paper submissions:** One sheet of 8.5x11 paper double sided, or two DISCONNECTED SHEETS with your name on EACH. Please: **NO STAPLES, NO PAPERCLIPS, NO TAPE, NO ATTACHMENT OF ANY KIND**. These damage the document scanner.

**Scanned submission:** When submitting by PDF upload: scan your pages legibly and **SCAN IN THE CORRECT ORDER**. Page 1 first, then Page 2, in the correct orientation. Failure to scan properly may result in zero credit, meaning you "use up" one of your five "drop the lowest grade" slots.

- (4 pts) Under what conditions does the compiler create a no-arg constructor for you?
- (4 pts) Under what conditions does the compiler NOT create a no-arg constructor for you?

4. (16 pts) Given the following code excerpts:

```
public class Person {  
    private String name;  
    public Person (String name) {this.name = name;}  
    public String getName() { return this.name;}  
}
```

Write a class for Student that extends Person. Include a private attribute perm of type int. Include a constructor with the following signature:

```
public Student(String name, int perm) { ...
```

Use the proper technique (pp. 250-257) for invoking the parent class constructor (with a parameter) to initialize the name attribute.

5. Based on what you learned from Chapter 9: Write a Java class that will compile and run (i.e. it needs a main() method) that has (at least) the following four variables: a, b, c, and d, each instance of which will have the properties indicated. The class doesn't have to do any useful work---it is only to illustrate that you understand these concepts.

- (5 pts) a should be a primitive variable that will be stored on the stack
- (5 pts) b should be an object reference that will be stored on the stack (note: the references is on the stack, even though the object it refers to will always be on the Heap in Java.)
- (5 pts) c should be a primitive variable that will always be stored on the heap.
- (5 pts) d should be an object reference that will always be stored on the heap (note: here I want the reference variable itself to be on the heap, not just the object it refers to.)