

W12:Homework:H07

From 56wiki

CS56—Advanced Applications Programming—W12

W12:Exams		W12:Homework				W12:Labs		W12:Calendar and Lecture Notes			W12:Syllabus			W12:Choice						
H00	H01	H02	H03	H04	H05	H06	H07	H08	H09	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20

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 (2,3,or 9)	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			

H07: Due Thu/Fri 01.26/01.27 in your ASSIGNED lab. Total Points: 50

MAY ONLY BE TURNED IN DURING THE CLASS INDICATED ABOVE, 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/W12:Syllabus> for more details.

Reading Assignment:

Read:

- HFJ:Chapter_9, **235** Life and Death of an Object (Constructors)
- If there are reading notes on the wiki, consult those too—sometimes they contain helpful hints.

(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 appear in Gauchospace),
- writing **either 2, 3 or 9** 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:	
email address:	@umail.ucsb.edu

(2) (5 pts) Under what conditions does the compiler create a no-arg constructor for you?

(3) (5 pts) Under what conditions does the compiler NOT create a no-arg constructor for you?

(4) (10 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) (20 pts) 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.

- a should be a primitive variable that will be stored on the stack
- 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.)
- c should be a primitive variable that will always be stored on the heap.
- 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.)

- This page was last modified on 13 January 2012, at 09:29.
- Content is available under Attribution-NonCommercial-ShareAlike 3.0 Unported.