

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

# 1

# H14

CS56 W16

## H14: Due Thursday, 01.28 in Lab

### More on Data Structures (HFJ Ch 16)

Assigned: Thu 01.21      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.)

There is no new reading assignment for this homework--it continues the study of Chapter 16 from HFJ.

During W12, the CS56 students were assigned to create homework questions from Chapter 16. The remainder of the questions on this assignment are based on questions they came up with.

Please:

- **No Staples.**
- **No Paperclips.**
- **No folded down corners.**

1. (10 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.

2. (Thanks to Vicente C.) List, Set and Map are fundamental concepts in the Collections API of Java. Briefly describe each of these (List, Set, and Map). For each, give an example of a programming situation where that one is more appropriate than the other two.

a. (10 pts) List

b. (10 pts) Set

c. (10 pts) Map

3. (10 pts) (Thanks to Raphael T.) Bob is trying to implement a car class with subclasses Honda, Ford, and Chevy. He wants to create a method which can polymorphically take an ArrayList of Car, or an ArrayList of any of its subclasses (e.g. ArrayList<Ford>, ArrayList<Honda>). He suggests the following method declaration:

```
public void takeCar(ArrayList<Car> list) { /* code goes here*/ }
```

What is wrong with Bob's logic and what would method declaration would correctly implement what he's trying to do?

# 2

# H14

CS56 W16