

CSU2500 Exam 1 HONORS SUPPLEMENT – Fall 2009

Name: _____

Student Id (last 4 digits): _____

Instructor's Name: _____

- This supplement to Exam 1 is intended for students enrolled in the Honors section of 2500.
- See the instructions on the regular exam.

Problem	Points	/out of
1		/ 5
2		/ 15
Total		/ 20

Good luck!

Problem 1 Recall Problem 4 from the regular portion of the exam. After your initial analysis of the submissions, you've decided that accepting only papers that got an A review eliminates far too many papers, so you decide instead to implement a more liberal acceptance policy: you'll rank the papers based on each paper's average reviews, and then accept the top papers until you have enough to make a symposium.

5 POINTS

Design a function that consumes a paper and computes the numeric average of its reviews. An A is worth 4 points; a B, 3; a C, 2; a D, 1; and a F is worth no points.

When formulating examples, you may use the names `g`, `b`, and `u`, as defined below, to save time writing:

```
(define g (make-paper "Good" 'A 'A 'A))
(define b (make-paper "Bad" 'C 'B 'B))
(define u (make-paper "Ugly" 'C 'D 'C))
```

[Here is some more space for the previous problem.]

Problem 2 You will solicit reviews from a number of your colleagues, who each are asked to review several papers. When they return the papers to you, they give the best papers first. You have to compute what the best papers are *overall*, so you're going to need to merge together the reviews each of the colleagues gives you.

Design a function, `merge`, that consumes two lists of papers, where the papers on each list appear in descending order of their average rating (so the best papers are first). The function produces a list that contains all of the given papers, ordered with the best (highest review average) papers first.

For example, if given these two lists:

```
(list (make-paper "Good" 'A 'A 'A)
      (make-paper "Ugly" 'C 'D 'C))
```

```
(list (make-paper "Bad" 'C 'B 'B))
```

the function should produce:

```
(list (make-paper "Good" 'A 'A 'A)
      (make-paper "Bad" 'C 'B 'B)
      (make-paper "Ugly" 'C 'D 'C))
```

[Here is some more space for the previous problem.]