

## COMP 170 Weeks 5-6 Homework

These are Chapter 5, Programming Projects 4, 9, and 5, plus #1 for extra credit.

4. (*using a Purchase class*) Write a program that uses the *Purchase* class in Listing 5.13 (found in the Sakai Week 5 **Source Code** folder) to set the following prices:

- Oranges: 10 for \$2.99
- Eggs: 12 for \$1.69
- Apples: 3 for \$1.00
- Watermelons: \$4.39 each
- Bagels: 6 for \$3.50

Then calculate the cost of each of the following five items and the total bill:

- 2 dozen oranges
- 3 dozen eggs
- 20 apples
- 2 watermelons
- 1 dozen bagels

9. (*updated Dog class*) Rewrite the *Dog* class given in Listing 5.1 by utilizing the information and encapsulation principles described in Section 5.2 of the text. The new version should include *accessor* and *mutator* methods. Also define an *equals()* method for the class that returns true if the dog's name, age, and breed match the same variables for the other object that is being compared. Include a *main()* method to test the functionality of the new *Dog* class.

10. (*Movie class*) Consider a class *Movie* that contains information about a movie. The class has the following attributes:

- The movie name
- The MPAA rating (e.g., G, PG, PG-13, R)
- The number of people who have rated this movie as a 1 (Terrible)
- The number of people who have rated this movie as a 2 (Bad)
- The number of people who have rated this movie as a 3 (OK)
- The number of people who have rated this movie as a 4 (Good)
- The number of people who have rated this movie as a 5 (Great)

Implement the class with *accessors* and *mutators* for the movie name and MPAA rating. Write a method *addRating()* that takes an integer as an input parameter. The method should verify that the parameter is a number between 1 and 5, and if so, increment by one the number of people rating the movie that matches the input parameter. For example, if 3 is the input parameter, then the number of people who rated the movie as a 3 should be incremented by one. Write another method, *getAverage()*, that returns the average value for all of the movie ratings based on those numbers.

Test the class by writing a *main()* method that creates at least two *Movie* objects, adds at least five ratings for each movie, and outputs the movie name, MPAA rating, and average rating for each *Movie* object.

***Hint:* There is an online videonote regarding a solution for Project 10.**

1. (*grading program using a StudentRecord class*) Write a grading program for an instructor whose course has the following policies: Two quizzes, each graded on the basis of 10 points, are given. One midterm exam and one final exam, each graded on the basis of 100 points, are given. The final exam counts for 50 percent of the grade, the midterm counts for 25 percent, and the two quizzes together count for a total of 25 percent. (Do not forget to normalize the quiz scores. They should be converted to percentages before they are averaged in.)

Any grade of 90 percent or more is an A, any grade between 80 and 89 percent is a B, any grade between 70 and 79 percent is a C, any grade between 60 and 69 percent is a D, and any grade below 60 percent is an F.

The program should read in the student's scores and display the student's record, which consists of two quiz scores, two exam scores, the student's total score for the entire course, and the final letter grade. The total score is a number in the range 0 to 100, which represents the weighted average of the student's work.

**Define and use a class for the student record.** The class should have instance variables for the quizzes, midterm, final, total score for the course, and final letter grade. The class should have input and output methods. The input method should not ask for the final numeric grade, nor should it ask for the final letter grade. The class should have methods to compute the overall numeric grade and the final letter grade. These last two methods will be *void* methods that set the appropriate instance variables. Remember, one method can call another method. If you prefer, you can define a single method that sets both the overall numeric score and the final letter grade, but if you do this, use a helping method. **Your program should use all the methods described here.** Your class should have a reasonable set of *accessor* and *mutator* methods, whether or not your program uses them. You may add other methods if you wish.

Note: to avoid problems with **Scanner** method interactions (*nextLine()* vs. any other *next()* method), you can use the **Keyboard** class methods I described previously.