CSCI 152 **Program 5**
Fall 2005 **Due: Friday, 9 Dec**

Emphasis on: **Classes, Dynamic Memory Allocation and Pointers**

The history teacher at your school needs help in grading a True/False test. The students' IDs and test answers are stored in a file. The first entry in the file contains the number of students in the class:

178

The second line in the file contains answers to the test in the form:

TFFTFFTTTTFFTFTFTFTT

Every other entry in the file is the student ID, followed by a blank, followed by the student's responses. For example, the entry:

AXZ5483 TFTFTFTT TFTFTFFTTFT

indicates that the student ID is AXZ5483 and the student's answer to question 1 was True, their answer to question 2 was false, and so on. This student did not answer question 9. The exam has 20 questions, and the class has more than 150 students. Each correct answer is awarded two points, each wrong answer gets -1 point, and no answer gets 0 points. Write a program that processes the test data. The output should be the student's ID, followed by the answers, followed by the test score, followed by the test grade. Assume the following grade scale: 90%-100%, A; 80%-89.99%, B; 70%-79.99%, C; 60%-69.99%, D; and 0%-59.99%, F. The maximum total possible score is 40 points (2 points X twenty questions), so calculate the numeric percentage grade assuming a 40 points as maximum.

Requirements:

1) You must dynamically allocate an array of pointers, based on the number of students in the file given as the first line in the data file.
2) You may want to start with your Grade class from Lab 5, modified appropriately for this task. You must create a Grade class to hold information about each student read in, such as their student ID, their scores, and their calculated percent and letter grades.
3) Also you must dynamically allocate new Grade objects to hold the information for each of the student grades.
4) After reading in the file and calculating the grades, display the results for each student in table form. You should display the Student's ID, raw score, percent grade and letter grade.
5) You should also compute an average raw score, and display the percent and letter grade this average represents for all of the students performances on the test.

Extra-Credit Opportunity

Create a function to sort your array of Grade object pointers, and sort the array based on the alphabetic order of the student's id. Perform the sort before display the resulting table of the student's performances. If you do this correctly, you should only be exchanging pointers in memory to do the sort, not whole Grade objects (which would be less efficient).