P R O G R A M G R A D I N G S H E E T

Sidney Sanders

Name:

1003

CS2020-section number

**Required portion:**

\_\_\_\_ [ 4 pts ] *main* function dynamically allocates memory for division and corporation structures. Data is read from the data file until end of file is reached. File object is passed to getDivision function. File is closed once end of file is reached.

\_\_\_\_ [ 4 pts ] **getDivision** function reads in the division name and four quarterly sales figures, uses pointer notation correctly to store the data in the Division structure.

\_\_\_\_ [ 4 pts ] **printDivision** function writes division information to screen, uses pointer notation correctly. Data aligned neatly in columns.

\_\_\_\_ [ 4 pts ] **addDivision** function adds quarterly figures to corporate totals, uses pointer notation correctly for both Corporation and Division structures.

\_\_\_\_ [ 4 pts ] **printCorpSummary** function writes correct corporate totals, averages and high/low quarters.

\_\_\_\_ [ 4 pts ] Program compiles and executes. Log file contains output from ALL requested commands (cat, g++, etc.). Log and source files have been collected from student’s directory on due date/time.

\_\_\_\_ [ 2 pts ] Program header and function headers contain required documentation items

\_\_\_\_ [ 2 pts ] Demonstrates good program style (i.e., meaningful identifier names, constant declarations, use of white space and in-line comments)

\_\_\_\_ [ 2 pts ] Answer the questions:

|  |  |
| --- | --- |
| Is there a limit to how many divisions a corporation can have? If yes, what is it? | No because the program creates memory for the divisions as it runs. |
| What are the advantages of dynamically allocating data for divisions and the corporation? | The advantage is that there can be endless amount of memory the program needs while the program runs. |

\_\_\_\_ / [ 30 pts ] **Total Points Earned**