CS 135
Assignment # *6*

Purpose: practice I/O files, prompt user for filename, verify that the file opened, and read data from a file.

Assignment:

The UNLV Computer Science Free Range Wild Animal Park has collected some data about feeding costs and stored that data into several text files. Each datafile contains the zoo's feeding totals for a day. In this assignment, you are going to add some functionalities to their zoo program to check those files. Here is the explanation of the files' structure.

- Each row of each file contains three columns: the animal name, the food that animal ate in pounds and the cost of that food respectively.

Part 1 – Opening and Closing the Datafile Overview

In the first part of your program, you need to prompt the user to obtain the file's name, open it up, and check whether the file has been opened successfully or not. The program should keep asking the user to enter the correct file's name if the file is not opened successfully. Here are the steps to follow to open a file:

1. Prompt the user to enter a filename and read the file name into a string.
2. Open up the file.
   - You are required to use the string from (1) to open the file up.
3. Check whether the file is opened successfully or not.
   - If the file has not been opened, output an error and go back to step 1.
4. Process the data from the file
5. Prompt the user to continue or exit
   - if the user chooses to continue, go back to step 1.
   - if the user chooses to quit, close the file and exit the program

Check Examples Below

Part 2 – Data-processing Overview

Process the data obtained from the file and output, to the display, the input file, and a table containing a header row and 3 columns: the animals name, the amount of food in pounds and the cost of that food in dollars. Following the table, print the total amount of food for the day in pounds and the total cost of the food for the day. From each file, use that data, to estimate the total amount of food the zoo will need for a month and the cost of that food for a month.

1. Print the input obtained from the file
2. Sum all of the animals' food obtained from the file
3. Sum all cost of all the animals' food, obtained from the file
4. Estimate monthly food need and cost based upon data obtained from file

Here are the steps to follow:

1. create the variables to hold the sums for the pounds of food and the cost
2. print an opening statement
3. prompt the user for the filename and verify successful opening of file, otherwise loop until this process is successful

4. print the user entered datafile
5. print a table header
6. read in and output each row of the file, calculating the sums until the end of file is hit
7. print the daily sums for the pounds of food and the cost
8. print the estimated monthly pounds of food and the estimated monthly cost
9. prompt the user to continue Y/y/N/n, looping until a proper response is entered
    1. if Y/y is entered, continue with (2)
    2. if N/n is entered, print a closing statement, close the input file and exit program

Assignment (Program) Technical Specifications:

1.) You must re-prompt the end user on bad file input.
2.) You must re-prompt the end user on an erroneous continue input response.
3.) According to animal safety protocols, Zoo food may be stored for a single day and re-served to the animals; therefore, animals may have food left over from the previous day. In the datafile, if food is served from the previous day, it is represented as a negative amount of food with a negative food cost.
    a.) Beware of this when calculating your totals.
4.) When you run the same file twice in a row, verify that you receive the same results.
5.) Separator lines for Welcome and Closing statements are 70 characters long.
6.) Separator lines for the Table Header and Feeding Totals are 50 characters long.
7.) Food pounds (lbs.) should be right aligned in the table and calculated to the tenths place.
8.) Food cost ($) should be right aligned in the table and calculated to the hundredths place.

An excellent website for looking up C++ functions is http://www.cplusplus.com/reference/ .

Submission:

Save your code as main.cpp. Do not ignore this step or save your file(s) with different names. Upload your main.cpp to CodeGrade before the due date/time.

Compiling:

Compile your program using the command: g++ -Wall -Werror -Wextra -pedantic main.cpp

Example Executions:

Example 1 – successful compile and execution

```
Karis-MBP:fileIO kari$ g++ -Wall -Wextra -pedantic -Werror main.cpp
Karis-MBP:fileIO kari$ ./a.out

**********************************************************************
Welcome the UNLV CS Free Range Wild Animal Park!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Please enter an input filename
***** zooData.txt

 Input file given: zooData.txt

+-------------------------------------------------+
 Animal           Food 'lbs.'      Cost '$'
+--------------+----------------+---------------+
 Giraffes                 9.8        $105.70
 Elephants               25.8        $320.26
 Lions                   39.8        $780.60
```

```
| Tigers          | 22.4 | $400.32 |
| Monkeys         | 12.4 | $145.33 |
| Llamas          |  6.7 | $45.11  |
| Butterflies     |  0.8 | $32.02  |
| Rabbits         |  4.4 | $48.83  |
| Penguins        |  8.9 | $280.65 |
| Ponies          | 15.3 | $76.89  |
| Bulls           | 35.2 | $658.57 |
+-----------------------------------------+

Feeding Totals
++++++++++++++++++++++++++++++++++++++++++++++++++
The total amount food the zoo needed today was: 181.6 lbs.
The estimated amount of food the zoo will need per month is: 5446.9 lbs.

The total cost of zoo food for today was: $2894.29
The estimated cost of food the zoo will need for a month is: $86828.75

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
Karis-MBP:fileIO kari$ g++ -Wall -Wextra -Werror -pedantic main.cpp
Karis-MBP:fileIO kari$ ./a.out

**********************************************************************
Welcome the UNLV CS Free Range Wild Animal Park!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Please enter an input filename
*****
```

Example 2 error opening input file on program start

```
Karis-MBP:fileIO kari$ ./a.out

**********************************************************************
Welcome the UNLV CS Free Range Wild Animal Park!
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

Please enter an input filename
***** a
ERROR opening the input file.
Please enter an input filename
***** b
ERROR opening the input file.
Please enter an input filename
***** c
ERROR opening the input file.
Please enter an input filename
*****
```

Example 3 error opening input file on continue processing datafiles

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                        Remember to always feed safely!
```

****

Would you like to process another datafile? Y/y/N/n
***** y

Please enter an input filename
***** a
ERROR opening the input file.
Please enter an input filename
***** b
ERROR opening the input file.
Please enter an input filename
***** c
ERROR opening the input file.
Please enter an input filename
*****

Example 3 successful compile, execution and process another datafile
Karis-MBP:fileIO kari$ g++ -Wall -Wextra -Werror -pedantic main.cpp
Karis-MBP:fileIO kari$ ./a.out

****
Welcome the UNLV CS Free Range Wild Animal Park!
++

Please enter an input filename
***** zooData_v2.txt

Input file given: zooData_v2.txt

Animal	Food 'lbs.'	Cost '$'
Giraffes	8.6	$85.38
Elephants	25.8	$320.26
Lions	20.2	$500.56
Monkeys	10.3	$105.33
Llamas	6.5	$43.21
Butterflies	0.5	$30.02
rabbits	5.2	$45.83

Feeding Totals
++
The total amount food the zoo needed today was: 77.2 lbs.
The estimated amount of food the zoo will need per month is: 2315.3 lbs.

The total cost of zoo food for today was: $1130.59
The estimated cost of food the zoo will need for a month is: $33917.81

++

Remember to always feed safely!
****

Would you like to process another datafile? Y/y/N/n
***** y

Please enter an input filename
***** zooData_v2.txt

Input file given: zooData_v2.txt

Animal	Food 'lbs.'	Cost '$'
Giraffes	8.6	$85.38
Elephants	25.8	$320.26
Lions	20.2	$500.56
Monkeys	10.3	$105.33
Llamas	6.5	$43.21
Butterflies	0.5	$30.02
rabbits	5.2	$45.83

Feeding Totals
++
The total amount food the zoo needed today was: 77.2 lbs.
The estimated amount of food the zoo will need per month is: 2315.3 lbs.

The total cost of zoo food for today was: $1130.59
The estimated cost of food the zoo will need for a month is: $33917.81

++
Remember to always feed safely!
****

Would you like to process another datafile? Y/y/N/n
***** n

Karis-MBP:fileIO kari$

Example 4 successful compile, execution and data file processing with food carried over from previous day
Karis-MBP:fileIO kari$ ./a.out

****
Welcome the UNLV CS Free Range Wild Animal Park!
++

Please enter an input filename
***** zooData_v3.txt

Input file given: zooData_v3.txt

```
+---------------------------------------------+
 Animal          Food 'lbs.'     Cost '$'
+--------------+----------------+--------------+
 Giraffes             10.0       $80.30
 Elephants            20.0      $300.10
 bob                  -3.0       $-100.00
+---------------------------------------------+

Feeding Totals
++++++++++++++++++++++++++++++++++++++++++++++++++++++
The total amount food the zoo needed today was: 30.0 lbs.
The estimated amount of food the zoo will need per month is: 900.0 lbs.

The total cost of zoo food for today was: $380.40
The estimated cost of food the zoo will need for a month is: $11412.00

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                                    Remember to always feed safely!
**********************************************************************************

Would you like to process another datafile? Y/y/N/n
*****
```

Example 5 successful compile, execution and invalid user input on continue processing datafiles

```
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                                    Remember to always feed safely!
**********************************************************************************

Would you like to process another datafile? Y/y/N/n
***** a

ERROR: Invalid entry. Please try again.
Would you like to process another datafile? Y/y/N/n
***** b

ERROR: Invalid entry. Please try again.
Would you like to process another datafile? Y/y/N/n
***** c

ERROR: Invalid entry. Please try again.
Would you like to process another datafile? Y/y/N/n
*****
```