Project Overview

- Write a C++ program to perform the following tasks
  - **Read** in grade information for three students from an input file
    - First name (multiple words possible)
    - Last name (multiple words possible)
    - Four test scores that are to be averaged
  - **Calculate** an average of the test scores for each student and an overall average for all three students
  - **Output** information is written to an output file and the format is as shown by running the sample solution:

    /home/work/cpe112/Executables/Project_06/Project_06_solution

- Use the comparison script to check Completed programs

  /home/work/cpe112data/Project_06/CompareSolution.bash Project_06.cpp

Project Constraints

- You may use any C++ programming constructs covered in Chapters 1-4 of your textbook or provided in the project handouts
- You may use the algorithm/Functional Decomposition provided as a guide when you implement your solution
- **The use of arrays and loops is prohibited**
- **Do not use global variables**

Project Constraints Continued

- *Your goal is to match the output style/format of your program to that produced by the provided sample solution.*

- *Your program should work for all provided input files and all conditions used in the compare solution script*

Hints

- When reading the input,
  - Use the getline function to read the first name and last name
  - Use the extraction operator to read in the scores (float values)
  - Don't forget about the new line character on the input stream
- Write a segment of code that will
  - Read in information for one person
  - Calculate the average for that person
  - Output the information for that person
  - Verify that this code works correctly, then duplicate it two times for the next two people
- Output the information as shown in the output file created by the sample solution
- Use two digits of precision for the numbers

- **Read the project description**

Functional Decomposition - 1

- Outline of main()
  - Header files needed: iostream, fstream, string, iomanip
  - Declare program variables – use floats for numerical values. Use string variables for the first and last names.
    - One set of variables is sufficient to handle all three people
    - Need a variable to hold the overall average of the three people
    - Need an input and output file stream variable and a string variable to hold the file name
  - Prompt for the name of the input file, read the name, echo print the file name and try to open the file provided by the user.
  - If the file did not open successfully,
    - Print out an error message (run the sample solution for content/format)
    - Terminate the program
    - See the project description for the necessary code.
  - Prompt for the name of the output file, read the name, echo print the file name and try to open the file provided by the user.
  - If the file did not open successfully
    - Print out an error message (run the sample solution for content/format)
    - Terminate the program
    - See the project description for the necessary code.

Functional Decomposition - 2

- Outline of main() - Continued
  - Read in the information line from the input file and write it to the output file
  - Output the column headers to the output file
  - Process the first person in the input file
    - Use the getline function to read the first name
    - Use the getline function to read the last name
    - Use the extraction operator to read in the four test scores
    - Average the four test scores
    - Add the average to a sum to use for the overall average
    - Output information
      - the first 9 characters of the last name (substr function)
      - The first 10 characters of the first name (substr function)
      - The average of the four test scores
  - Process the second person in the input file
  - Process the third person in the input file

Functional Decomposition - 3

- Outline of main() - Continued
  - Output the overall average of the three people
  - Close the input and output file streams

Note: When processing the second and third person, the last input operation performed for the previous person is an extraction. The first input operation performed for the second and third person uses the getline function.

Output Formatting

- Formatting of the output
  - Create the column headers with –'s underneath the words
  - Last name is in a field width of 12 left justified– output is the first nine characters of a persons last name
  - First name is in a field width of 12 left justified– output is the first ten characters of a persons first name
  - Average follows immediately after the first name field – no field width is specified
  - Place the dashes under the column headers after the information for the third person
  - Output of the overall average line
    - Output Overall for the last name column,
    - Average for the first name column and
    - Then the numeric overall average value