**CS 37 Program Sets**

**Spring 2021 Semester**

**For full credit on each program, please paste your output to the bottom of your program listing as a comment and make sure to include your full name at the top of the program as a comment. Also please make sure to format your output properly. For example, include appropriate spacing and blank lines and format money amounts to two decimal places after the decimal point.**

All homework programs will need to be submitted through Canvas and **late homework cannot be accepted for any reason.**

**Program 1:**  You need a program to maintain information about pay raises for employees. Each employee's data include their name, their department, their current yearly income, and their raise percentage. Assume that the following data is to be used:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Department | Current Yearly Income | Raise Percentage |
| James, Kobe | Accounting | $66111.54 | 6.6% |
| Walton, Lebron | Marketing | $89224.17 | 4.8% |
| Rondo, Luke | Sales | $75123.25 | 7.4% |
| Bryant, Kyle | Sales | $69644.44 | 5.5% |
| Kuzma, Rajon | Marketing | $96222.77 | 3.9% |

Write a C++ program which **uses appropriate functions and parameter passing** that will perform the following tasks:

**Task 1:** Allow the user to enter in the data for each employee. (The user will not enter the $ or % symbols.)

**Task 2:** For each employee, calculate the Raise Amount and the New Pay Amount.

Raise Amount = Current Yearly Income \* Raise Percentage / 100

New Pay Amount = Current Yearly Income + Raise Amount

**Task 3:** For each employee, print out of their Name, Department, Current Yearly Income, Raise Percentage, Raise Amount, and New Pay Amount.

**Task 4:** Calculate and print the total of the current yearly incomes, the total of the raise amounts, and the total of the new pay amounts for the five employees.

**Task 5:** Print the average of the Raise Amounts along with the largest and smallest Raise Amounts.

**Program 2:** Modify arrtemp.cpp by removing the preassignment of the data for each array and by adding five new function templates. The first new function template should allow the user to enter the array data from the keyboard. The second new function template should print the average of the array with the two largest values excluded from the average calculation. The third new function template should sort the array data in ascending order. The fourth new function template should save the array data to a text file. The fifth new function template should retrieve the array data from the text file. Output should include the average of each numeric array with the two largest values excluded and all three arrays being printed out in ascending order twice, once before the text file is saved and once after the array is retrieved from the text file.

(Warning - make sure to include the line **template <class T>** before each function template.) **Each array should be saved to a separate text file**. You will also need to show all three text files which you can do from any text editor when submitting this program.

**Program 3:** Create a class that has the following methods for working with a 10 element integer array:

1. A default constructor which sets all the array elements to 0.
2. Pick 10 random integers from 1 to 100 to be loaded into the array.
3. Print the array.
4. Sort the array in ascending order and print the sorted array.
5. Sort the array in descending order and print the sorted array.
6. Find and print the average of the array with the two largest values excluded from the average calculation.
7. Save the array to a text file.
8. Retrieve the array from the text file and print the array.
9. Add the corresponding elements of two arrays and store the sums into a separate object.
10. Subtract the corresponding elements of two arrays and store the differences into a separate object.
11. Reverse the order of the array elements.

Your array should be declared privately in the class. Your main should call each of the class methods and output should be shown for methods c, d, e, f and h. In addition, the arrays found in methods i and j should be printed. Lastly, each array should be printed after method k is called. (**Please do not overload the operators for this program.**)

**Program 4:** Rewrite Program 3 where the following operators are overloaded:

a. << to print an array

b. >> to allow the user to enter the values for an array.

c. + to add the corresponding elements of two arrays and store the sums into a separate object.

d. - to subtract the corresponding elements of two arrays and store the differences into a separate object.

e. > returns true if the average of the first array with the two largest values excluded from the average calculation is greater than the average of the second array with the two largest values excluded from the average calculation.

f. <= returns true if the average of the first array with the two largest values excluded from the average calculation is less than or equal to the average of the second array with the two largest values excluded from the average calculation.

No saving or retrieving of a text file is needed.

Also overload the predecrement and postdecrement operators -- which will reverse the order of the array elements.

**Final Exam Program:**

This program should calculate final percentages and grades for the following types of courses using the following grading scale:

**Final Percentage** **Grade**

90% or more A

80% to less than 90% B

70% to less than 80% C

60% to less than 70% D

Less than 60% F

Each course grade is based on the average of the lab percentage and classroom percentage. For example, if the lab percentage for a student is 65% and the classroom percentage is 85%, then the final percentage is 75% which is a final grade of C.

**Course Types**

1. Psychology Courses: Lab percentage is based on 3 labs worth up to 100 points each, and the classroom percentage is based on 4 problem sets worth up to 25 points each, and 2 exams worth up to 150 points each.

2. Biology Courses: Lab percentage is based on 3 labs worth up to 100 points each where the lowest lab score is dropped, and the classroom percentage is based on 3 quizzes worth up to 50 points each, and 4 exams worth up to 100 points each where the lowest quiz score and the lowest exam score are dropped.

3. Computer Science Courses: Lab percentage is based on 4 program sets worth up to 25 points each, and the classroom percentage is based on 2 exams worth up to 100 points each, and a final exam worth up to 100 points. If the final exam score is higher than the lowest exam score, then the lowest exam score is replaced by the final exam score. If the final exam score is under 50 points, the student automatically fails the course.

4. A Course Type of your choice: Please include documentation (comment statements) stating how the lab percentage and the classroom percentage are calculated for your course.

For each course, print out the course number, course name, student name, lab percentage, classroom percentage, final percentage, and final grade. Please make sure to include output for each of the four course types.

Your program should use virtual functions including at least one pure virtual function with polymorphism. Your program should have one class for each course type along with a base-class that contains the appropriate data members and methods.