CS 150 Lecture 2 Exercises

Complete each of the exercises below. Upload them to Canvas to get credit for this assignment.

| *Exercise 1 - a screenshot of your test results here* |
| --- |

| *Exercise 1 - a screenshot of your source code here* |
| --- |

Exercise 1 - Design, Code & Run a C++ Program  
In **Homework 0** you learned how to compile, run and test C++ programs. Use the **CS50 IDE** to design, code and test the problem shown here.

Here is the problem description:

*Workers at a particular company have won a* ***7.6%*** *pay increase retroactive for six months. Write a program that takes an employee’s previous annual salary as input and outputs the amount of retroactive pay due the employee, the new annual salary, and the new monthly salary. Use a variable declaration with the modifier* ***const*** *to express the pay increase.*

Here is what the program should look like when it runs:

Enter current annual salary: 10000

New annual, monthly and retroactive salary: [10760.00, 896.67, 380.00]

* Design the algorithm:
  + For input, you’ll need the original salary.
  + For output you’ll need to calculate the new annual salary.
  + With the new salary, you can calculate the new monthly salary
  + To calculate the retroactive pay, subtract the old salary from the new. This would be the retroactive pay for an entire year. You want only the pay for six months.
* **Code the solution:**
  + Update the file with your name (2 places), the date and the assignment. Use your Canvas/occ-mail name (like **sgilbert**) for the **STUDENT** name
  + Update the comment for the **run()** method using information in this handout.
  + Add the output statement inside the **run()** function to print the prompt.
  + Compile and run using **make** to see that you have the mechanics working.
  + Create a **double** variable to hold the user’s original salary. Use the **cin** input object to read the input after the prompt. Use **make run** to see that it works.
  + Add a **const** **double** definition for the amount of increase (**.076**).
  + In the **run()** function, create variables for the three output items: **annualSalary**, **monthlySalary** and **retroactivePay**. Calculate and assign the values to these three variables using the formula described above.
  + Print the output prompt and the output line as shown in the example above. Build and run. Test with **10000** as the input.
  + To make the decimals appear correctly for all output **#include <iomanip>** and add the following line to your program before doing output.
    - **cout << fixed << setprecision(2);**
    - You only need this once in your program.
* **Testing the Program**
  + Test your program for at least 4 additional different initial salary values in addition to the values that I've supplied. Calculate the expected values for each of the four inputs. Then use **make** **stest** to check your results. If you don’t have 100%, then fix the problem, rebuild the project and test once again.