ACMS 20210 Assignment 1

Due: 11:59 PM on Monday, September 3

Submit the C++ programs below using Sakai. If you are compiling from the command line on the CRC or another Linux system, I suggest using the command

```
g++ your_program_name.cpp -std=c++1y -o your_executable_name
```

This tells the compiler (called g++) to compile your .cpp source file into an executable file. To run this file from the command line, type

```
./your_executable_name
```

If you do not yet know how to use the CRC and do not have a compiler on your local machine, I suggest using the website *cpp.sh* to test your code (be sure to save a copy on your local machine, though!). You must submit your code for each problem in a separate .cpp file (make certain to submit the .cpp file, **not** the executable files).

Use the following naming scheme for submitting your solutions on Sakai: For exercise n on homework m, the name of your submission should be hw_m_ex_n.cpp. As an example, for homework 1 exercise 3, the name of your submission should be: hw_1_ex_3.cpp. If you do not adhere to this naming scheme, you may lose some credit.

1. Modify the broken code in the file hw_1_ex_1_broken_code.cpp on Sakai so that it compiles **and** produces nice looking output. You should look for things like missing #include, missing quotes or semicolons, lack of types specified, etc.

2. Write a program that requests the user to input two integers, a and b, and then prints their sum, difference and product (with appropriate messages to the user telling them which quantities are being printed).

3. In this exercise, we will simulate computing the total of a grocery bill at a small fruit stand. The stand sells apples, bunches of bananas, oranges and lemons. Each apple costs $1.00, each bunch of bananas costs $2.75, each orange costs $1.25 and each lemon costs $1.75. Write a program that prompts the user to enter the number of each item (apples, bunches of bananas, oranges, and lemons) and computes and reports the total number of items and the total grocery bill in dollars. Make certain that your prompts are reasonably worded. After each prompt, display the total bill for that item, and after all four prompts, display the overall totals. Executing the program might look something like the following:

   Please enter the number of apples: 2
   The total bill for 2 apples is: $2
   Please enter the number of bunches of bananas: 4
   The total bill for 4 bunches of bananas is: $11
   Please enter the number of oranges: 2
   The total bill for 2 oranges is: $2.5
   Please enter the number of lemons: 0
   The total bill for 0 lemons is: $0

   The total number of items purchased is: 8
   The overall total bill is: $15.5

   You do **not** have to account for grammar (e.g. "1 oranges" is fine), the number of decimal spaces after the decimal point, or for the possibility that the user enters a negative number (or invalid input) for the number of a given fruit. You should give your variables reasonable names that reflect their meaning (e.g. num_apples to store the number of apples) and your variables should have reasonable types.