CS 121 SI Worksheet - Week 3 (9/10)

**iomanip:**

Short for IO manipulation, the iomanip library allows us to make text look pretty. Although there are a good amount of functions within the library, the following are the most commonly used:

- **setfill(char ch):** Set a character to fill when needed (pretty much just for setw). Default char is space.

- **setprecision(int x):** Set the amount of digits you wish numbers outputted go to (automatic rounding)

- **setw(int x):** Set field width to be used when outputting

There are a few manipulators (not included in iomanip) that are commonly used with the above as well:

- **fixed**: To be used with **setprecision**; makes **setprecision** round after the decimal

- **left/right:** Left/right justification

**Example nonsensical code segment (applying above manipulators):**

int num1 = 12, num2 = 4495, num3 = -1455;

const double PI = 3.141592;

const double EULER\_NUM = 2.71828;

cout << setfill('x');

cout << EULER\_NUM << endl;

cout << setw(6) << num1 << " " << setw(3) << num3 << " " << num2 << endl;

cout << setw(10) << left << setprecision(5) << PI << endl;

cout << fixed << PI << endl;

cout << setfill('y') << setw(5) << EULER\_NUM << setw(12);

cout << right << "hello" << endl;

**Output:**

2.71828

xxxx12 -1455 4495

3.1416xxxx

3.14159

2.71828yyyyyyyhello (seven y's)

**Exercise with iomanip:**

- Write a code segment (the output section only) to do the following: Use the following three variables to make a print out as shown afterwards. Assume each variable is initialized and good for use.

const int ITEM\_CAP; //limit on how many items there can be in total

string name\_list[ITEM\_CAP]; //list of item names

double price\_list[ITEM\_CAP]; //list of prices

Output (one possible example):

Item Price

-------------------

Toy car $2.00

Doll $3.00

Pillow $5.00

INFO: The entries are left-justified and you must include the '$' yourself when outputting.

HINTS: After printing out the header, use a for loop to loop through the arrays.

NOTE (EXTRA): name\_list and price\_list are parallel arrays. That means name\_list[i] and price\_list[i] correspond to one another (e.g. if the first item was a toy car and its price was two dollars, then we would have name\_list[0] = "Toy car" and price\_list[0] = 2.00).

**BONUS:** Assume the previous example had a partially-filled array (i.e. there are less than ITEM\_CAP amount of items in the list). What would we have to change in the previous loop so it would work? Use the following variable for the "true" item count:

int item\_count; //actual number of items in lists

Miscellaneous questions:

1. What's the difference between the following two lines of code? Is there a way we can make the first equal to the second (without changing 14 -> 14.0 and 5 -> 5.0)? If no, explain why. Otherwise, show a way that's possible.

cout << 14/5 << endl;

cout << 14.0/5.0 << endl;

2. Why do you think programmers typically type out constant variables in all capitals? Think about how they look compared to other variables made.

3. List at least three reasons on why your code's syntax must be clean, readable, and consistent (in reference to typing and naming style).

4. What is the syntax for assigning the following formulas to a double called fx? Assume all variables needed are declared and initialized, and use cmath functions if needed. **HINT:** Work inwards to outwards when writing out the nastier formulas.