iomanip

Making output look good
Goals

- By the end of this lesson you will be able to:
  - Utilize manipulators from `ios`
  - Utilize manipulators from `iomanip`
  - Write programs that produce better looking output
Why format output?

• When using text, formatting output can be very useful for a variety of reasons:
  – To make the program’s output more visually appealing
  – To make the program’s output easier to read
  – To make the program’s output have the appropriate level of accuracy with respect to how many digits are shown
Manipulators can be used in output statements involving `cout` or an `ofstream` variable.

There are many useful manipulators and they exist primarily in two header files: `ios` and `iomanip`.

You do not need to include `ios` because you inherit it by including `iostream`, but you do need to include `iomanip` if you use manipulators from that file.
Manipulators

- Most manipulators are a toggle and are active for the entire program until you disable them
- You can combine manipulators with anything else you are outputting or have them alone on a line by themselves
ios manipulators

- **fixed** – Output floating-point values with the number of digits after the decimal to the set precision (default 6)
  - To disable, use `cout.unsetf(ios::fixed);`
- **showpoint** – Output a decimal point for all floating-point values and match the set precision for the entire number
  - To disable, use `noshowpoint` modifier
- **scientific** – Output floating-point numbers in scientific notation
- **left and right** – when using `setw` from `iomanip`, set the alignment
iomanip manipulators

• setw(int value) – output the next expression such that it takes int value columns

• setfill(char value) – fill extra space due to setw with a specific character (default is the blank space)

• setprecision(int value) – specific precision for floating-point values (use in conjunction with ios manipulators)
Manipulator notes

- There are many other manipulators that you can find by checking a reference, for example:
  - Automatic conversion to other numeral bases (hexadecimal, octal)
  - Display of true/false instead of 1/0 when outputting a `bool`
  - Place blank spaces between a negative sign and the value
  - Output values representing time & money in pre-set ways
• Write a program that will take in 3 floating-point values from the user, then output them as below:

<table>
<thead>
<tr>
<th>Value entered</th>
<th>Net sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.00000000</td>
<td>6.00000000</td>
</tr>
<tr>
<td>3.14159000</td>
<td>9.14159000</td>
</tr>
<tr>
<td>2.50000000</td>
<td>11.64159000</td>
</tr>
</tbody>
</table>

• Examine and test other manipulators by looking at references for **ios** and **iomanip** manipulators