Basic Input/Output formatting

In order to use the formatting facilities described below, you need to include *iomanip*, in addition to *iostream*:

```
#include <iostream>
#include <iomanip>

using namespace std;

int main() {

  //YOUR PROGRAM HERE
return 0;
}
```

1- *setw*: Adjusts the number of positions (i.e. length of the field) reserved for your output. Only affects the next value sent to the output.

Example:

#include <iostream>

#include <iomanip>

using namespace std;

```
int main()
{
cout << '(' << "Two words" << ')' << endl;
cout << '(' << setw(20) << "Two words"
<< ')' << endl;
cout << '(' << setw(30) << "Two words"
<< ')' << endl;
return 0;
}
```

If the data needs more space than specified in *setw*, the space needed will be used instead of the width specified in *setw*.

left and *right:* Used to left justify or right justify the output.

Example:

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
```

```
{
cout << '(' << "Two words" << ')' <<
endl;
cout << left; //following output is left
                //justified
cout << '(' << setw(20) << "Two words"
<< ')' << endl;
cout << right; //following output is
                //right justified
cout << '(' << setw(30) << "Two
words" << ')' << endl;
return 0;
}
```

The output is :

```
(Two words)
(Two words           )
(                     Two words)
```

2- *setfill*: sets the fill character so that it fills any spaces with that character if the output is shorter than the width specified by *setw*.

For example, in the code above, adding *setfill('*')*

```
#include   <iostream>
#include <iomanip>
using namespace std;

int    main()
{
cout <<   '('   <<   "Two words"   <<   ')'   <<
endl;
cout <<   left;  //following      output     is     left
                 //justified
cout << setfill('*') << '('   <<   setw(20)   <<
"Two words"   <<   ')'   <<   endl;
cout <<   right;      //following      output     is
                      //right     justified
cout <<   '('   <<   setw(30)   <<   "Two words"<<
')'   <<   endl;
return      0;
```

```
}
```

results in

```
(Two   words)
(Two   words**********)
(********************Two    words)
```

3- Floating Point Numbers (i.e. decimals)

We will cover only decimal output in the decimal point format (not scientific notation, etc.)

Precision: basically the number of digits after the decimal number.

Example:

```
#include <iostream>
#include <iomanip>
using namespace std;
int     main()
{
cout   <<     setprecision   (3)     <<     1.0/3.0<<     endl;

cout   <<     setprecision   (9)     <<     1.0/3.0<<     endl;

return 0;
}
```

More on manipulators can be found at

https://www.cs.mtsu.edu/~rwsmith/2170/examples/formattingOutput.html

Use the Quick Link on that page to jump to the sections on *setw*, *setfill*, *right*, *left*, and *setprecision*.