THE UNIVERSITY OF VERMONT
COLLEGE OF ENGINEERING &
MATHEMATICAL SCIENCES

OUTPUT MANIPULATION

CS 124 / Department of Computer Science

Review: Lecturer class

In our previous module, Introduction to C++, you saw some output manipulators including:

```
std::left
std::setfill()
std::setw()
std::boolalpha
```

In this video, we'll review these and other output manipulators.

Output manipulation

- Set the width of columns
- Control the display of floating-point numbers
- Align output
- Make output prettier and more readable

iomanip and iostream

When we wish to modify our output stream with manipulators that take parameters, first we include the `iomanip` library, with the appropriate preprocessor directive:

```
#include <iomanip>
```

Manipulators that do not take parameters are included with `iostream`.

Manipulators for all data types

Manipulations you can use with all data types:

`std::setw(n)` — whatever is printed next will take at least *n* characters, aligned right

`std::setfill(n)` — specifies the padding character

`std::left` — aligns left

`std::right` — aligns right

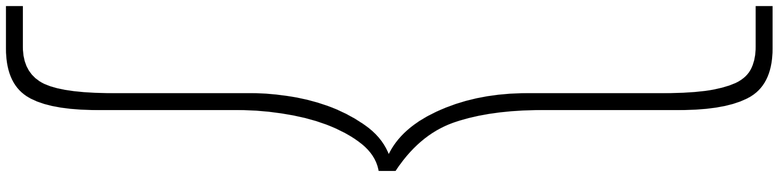
std::setw()

```
std::cout << "foo! bar!" << std::endl;

std::cout << std::setw(20) << "foo! bar!" << std::endl;
```

Prints...

```
foo! bar!
            foo! bar!
```



20 characters

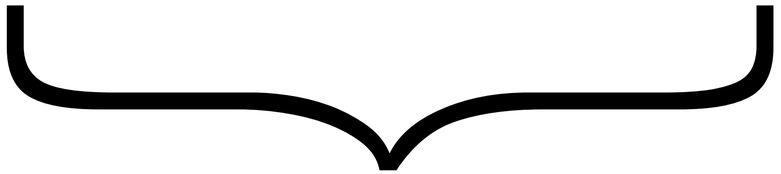
std::setfill()

```
std::cout << std::setfill('.');

std::cout << std::setw(20) << "foo! bar!" << std::endl;
```

Prints...

```
............foo! bar!
```



20 characters

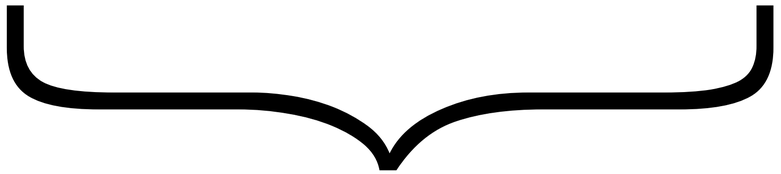
std::setfill()

```
std::cout << std::setfill('.');

std::cout << std::setw(20) << "foo! bar!" << std::endl;
```

Prints...

```
...........foo! bar!
```



20 characters

Remember:

`std::setfill()` takes a character as a parameter and characters are denoted with single quotation marks.

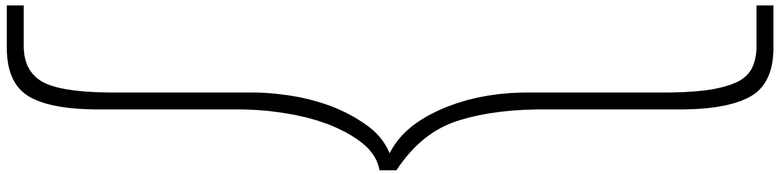
std::setfill()

```
std::cout << std::setfill(' ');

std::cout << std::setw(20) << "foo! bar!" << std::endl;
```

Prints...

```
          foo! bar!
```



20 characters

std::left

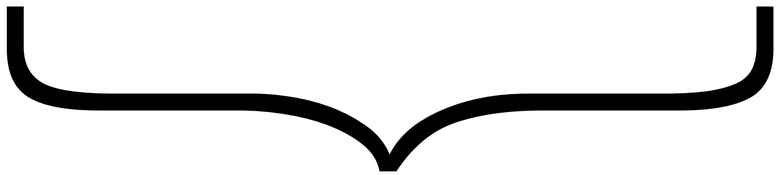
```
std::cout << std::left;

std::cout << std::setfill('.');

std::cout << std::setw(20) << "foo! bar!" << std::endl;
```

Prints...

```
foo! bar!...........
```



20 characters

std::right

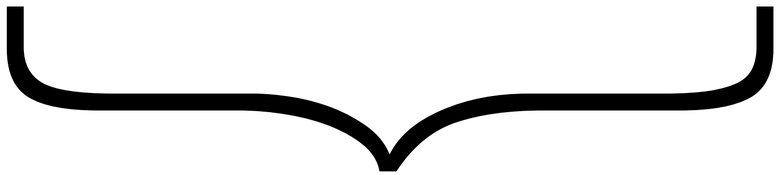
```
std::cout << std::right;

std::cout << std::setfill('.');

std::cout << std::setw(20) << "foo! bar!" << std::endl;
```

Prints...

```
...........foo! bar!
```



20 characters

Manipulators for floating point numbers

Manipulations you can use with floating point numbers:

`std::fixed` — fixes the decimal place

`std::setprecision(n)` — limits all doubles to print *n* digits after the decimal place

`std::showpoint` — display decimal point

`std::noshowpoint` — do not display decimal point

`std::scientific` — display in scientific format

std::noshowpoint and std::showpoint

```
double five = 5.0;

std::cout << std::noshowpoint << five << std::endl;

std::cout << std::showpoint << five << std::endl;
```

Prints...

```
5

5.00000
```

std::noshowpoint and std::showpoint

```
double pi = 3.14159;

std::cout << std::noshowpoint << pi << std::endl;
```

Prints...

```
3.14159
```

Note:

`std::noshowpoint` only works with floating point numbers that have zero to the right of the decimal point.

std::fixed and std::setprecision()

```
double pi = 3.141592;
std::cout << std::fixed << pi << std::endl;
std::cout << std::setprecision(2) << pi << std::endl;
std::cout << std::setprecision(4) << pi << std::endl;
```

Prints...

```
3.141592

3.14

3.1416
```

std::fixed and std::setprecision

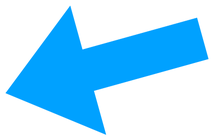
```
double pi = 3.141592;
std::cout << std::fixed << pi << std::endl;
std::cout << std::setprecision(2) << pi << std::endl;
std::cout << std::setprecision(4) << pi << std::endl;
```

Prints...

```
3.141592

3.14

3.1416
```



Caveat

```
double five = 5.0;

std::cout << std::fixed << pi << std::endl;

std::cout << std::noshowpoint << five << std::endl;
```

Prints...

```
5.000000
```

`std::fixed` takes precedence

std::scientific

```
double c = 299792458;  // speed of light in m / s


std::cout << std::scientific
          << std::setprecision(4) << c << std::endl;
```

Prints...

```
2.9979e+08
```

std::boolalpha and std::noboolalpha

```
std::cout << std::boolalpha << true << std::endl;
std::cout << false << std::endl;
std::cout << std::noboolalpha << true << std::endl;
std::cout << false << std::endl;
```

Prints...

```
true
false
1
0
```

Which libraries are needed?

#include <iostream>	#include <iomanip>
std::left	std::setfill()
std::right	std::setw()
std::fixed	std::setprecision()
std::scientific	
std::showpoint	
std::noshowpoint	
std::boolalpha	
std::noboolalpha	