Basic Formatting (<iostream>)

Special Characters

- **endl**: Newline
- **\n**: Newline
- **\t**: Tab
- **\"**: Double quote
- \\: Backslash

```
cout << "Newline: " << endl;

cout << "Newline: \n";

cout << "Tab:\tHere" << endl;

cout << "Quote: \"" << endl;

cout << "Backslash: \\" << endl;
```

cout.width() and cout.fill()

- width(): Sets field width
- fill(): Sets fill character for padding.

```
cout.width(5);
cout.fill('*');
cout << "1" << endl; // Outputs "****1"
```

cout.precision()

- precision(): Sets the precision for subsequent floating-point outputs.

```
cout.precision(3);
cout << 3.14159 << endl; // Outputs "3.14"
```

Advanced Formatting (<iomanip>)

setw()

- setw(): Sets the field width
- setfill(): Sets the fill character
- setprecision(): Sets the precision for the next floating-point output only.
- left: Aligns output to the left within the field width. For
- right**:** Aligns output to the right within the field width (default).

```
cout << setw(5) << "1" << endl;  // Outputs "    1"

cout << setw(5) << setfill('*') << "1" << endl;  // Outputs "****1"

cout << setprecision(3) << 3.14159 << endl;  // Outputs "3.14"

cout << setw(5) << left << "1" << "next" << endl;  // Outputs "1    next"

cout << setw(5) << right << "1" << "next" << endl;  // Outputs "    1next"
```

cout.width() vs setw()

- cout.width(5) and setw(5) are functionally equivalent in that they both set the minimum field width for the next output operation only. After that, the width setting is reset.

cout.fill() vs setfill()

- cout.fill('*') and setfill('*') are also similar in their function. x

cout.precision() vs setprecision()

- The difference is that setprecision() are the same. They both persists until changed

Number and Symbol Formatting (<iomanip>)

showpoint and noshowpoint

- showpoint: Forces the display of the decimal point.
- noshowpoint: Removes the decimal point if there are no decimal digits.

```
cout << showpoint << 3.0 << endl;  // Outputs "3.00000"
cout << noshowpoint << 3.0 << endl;  // Outputs "3"
```

showpos and noshowpos

- showpos: Shows the + sign for positive numbers.
- noshowpos: Does not show the + sign for positive numbers (default).

```
cout << showpos << 3 << endl;  // Outputs "+3"
cout << noshowpos << 3 << endl;  // Outputs "3"
```

oct, dec, hex, uppercase

- oct: Outputs integers in octal format.
- dec: Outputs integers in decimal format (default).
- hex: Outputs integers in hexadecimal format.
- uppercase: Makes all letters uppercase in the next output.

```
cout << oct << 15 << endl;  // Outputs "17"
cout << dec << 15 << endl;  // Outputs "15"
cout << hex << 15 << endl;  // Outputs "f"
cout << uppercase << hex << 15 << endl;  // Outputs "F"
```