Formatted output

- ***fixed*** and ***showpoint***

  - by default, small numbers are displayed using fixed format.
    large numbers are displayed in scientific format

  - **fixed** manipulator allows decimal, not scientific notation be used.
  - **showpoint** manipulator allows decimal point to be included in the output, even for values with 0 as fractional part.
  - Both are defined in <iostream>

  Example:

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

int main()
{
     float  value1 = 1.;
     float  value2 = 1.234;
     float  value3 = 1.2345678;
     float  value4 = 1234567.875;

     // print values without any formatting
     cout << value1 << endl << value2 << endl;
     cout << value3 << endl << value4 << endl;

     // print values to show in non-scientific form, and decimal point shown for
     // floating values
     cout << fixed ;
     cout << showpoint;
     cout << value1 << endl << value2 << endl;
     cout << value3 << endl << value4 << endl;

     return 0;
}
```

- **setprecision(n)**

  - **Defined in <iomanip>**
  - If **fixed** has already been specified, argument **n** determines the number of places displayed after the decimal point for floating point values
  - Remains in effect until explicitly changed by another call to **setprecision**
  - Value is **rounded** if necessary

  Example

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

int main()
{
     float  value1 = 1.;
     float  value2 = 1.234;
```

```
        float  value3 = 1.2345678;
        float  value4 = 1234567.875;

        // print values without any formatting
        cout << value1 << endl << value2 << endl;
        cout << value3 << endl << value4 << endl;

        // print values to show in non-scientific form, and decimal point shown for
        // floating values
        // demonstrate setprecision formatting command
        cout << fixed;
        cout<< showpoint;
        cout << setprecision(2);
        cout << value1 << endl << value2 << endl;
        cout << value3 << endl ;
        cout << setprecision(1) << value4 << endl;

        return 0;
    }
```

- **setw**(width)
  setw : control the number of character positions the next data item should occupy when it is output
  width : field width specification
    - apply to numbers and strings, not char type data
    - default to be **right justified**
    - empty spaces are default to be filled w/ ' ' (blank space)
    - if size of value (i.e., number of digits in value) > setw width, setw is ignored
    - setw only affects the next item displayed, have to use setw for every output value.

  Example:

```
(1) int NumStudents = 26;
    cout << "Number of students in the class is "
         << setw(5) << NumStudents << endl;

(2)  cout << "Number of students in section " << setw(8)
          <<  "Sec5"  << "  is  " << NumStudents << endl;

 (3) float  balance = 1300.87;
     cout << fixed;
     cout << showpoint;
     cout << setprecision(1);
     cout << "The current account balance is $" << setw(8) << balance << endl;

(4)  int  myNumber   =  123 ;
     int  yourNumber  =  5 ;
     cout  <<   setw ( 10 )    <<  "Mine"
           <<   setw ( 10 )    <<  "Yours"        <<   endl;
           <<   setw ( 10 )    <<  myNumber
           <<   setw ( 10 )    <<  yourNumber   <<   endl ;
```

- **left and right justification** : justification remain valid until it is reset.
  cout << left; or cout << right;

Example

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

int main ()
{
          // print the heading
          cout << endl << endl;
          cout << left;
          cout << setw(15) << "School Number"
               << setw(15) << "School Name" << endl << endl;
          cout << right;

          // print the 1st school
          cout << setw(5) << 10 << setw(10) << " ";
          cout << left;
          cout << setw(15) << "MTSU" << endl;

          // print the 2nd school
          cout << right;
          cout << setw(5) << 5 << setw(10) << " ";
          cout << left;
          cout<< setw(15) << "UT" << endl;

          // print the 3rd school
          cout <<  right;
          cout << setw(5) << 32 << setw(10) << " ";
          cout << left;
          cout << setw(15) << "Vanderbilt" << endl;
          cout << endl << endl;

        return 0;

}
```