How to control layout and formatting on the console

CS 10A – OUTPUT FORMATTING

Allen Zhao - SRJC CS 10A

Formatting the Output

- Normally, we have to be very specific on where and how much space we need to use on the console whenever we handle our output.
- There are several commands and alternative syntax we can use in C++ to get better control over how our text looks, whether its numbers or characters.
- The primary application is specifying how many decimal places of accuracy to display with numerical outputs.
- Other applications include faking graphics, such as creating a map for text-based adventure games.

IO Manipulation Library

- <iomanip> is the Input/Output Manipulation library
- This library has a number of formatting functions, but the following two are by far the most useful:
  - setw(x) – sets the width of a column to x spaces
  - setprecision(x) – set a number to display with x significant figures
- These modifiers are available, but are more limited in use:
  - fixed – fixed point notation, use in conjunction with setprecision() to set number of decimal places to output instead of significant figures
  - showpoint – similar to fixed, but used solely for padding excess 0s when not enough decimal places exist
  - defaultfloat – the opposite of fixed, this is the default for setprecision()
  - left – left justification of text
  - right – right justification of text

Setting the Width

Program

```
#include <iomanip>
int main()
{
        cout << setw(6) << “Hi” << endl;
        /* The next element in cout is exactly 6
         * characters in width, including whatever
         * text follows it. 4 spaces will precede “Hi”.
         * Text by default is right-justified
         */
        cout << setw(9) << “Hello” << endl
             << setw(9) << “World\n”;
        // Escape characters count as characters!
}
```

Console

- ```
  ./a.exe
    Hi
    Hello
  World
  ```
  ```

Setting the Precision

Program

```
#include <iomanip>
const double pi = 3.14159;
const double e = 2.71828;
int main()
{
        cout << setprecision(3) << pi << endl;
        cout << setprecision(8) << e << endl;
        // Set the significant figures to display
        // You can't display more decimals
        // than what the variable currently has
        return 0;
}
```

Console

- ./a.exe

```
3.14
2.71828
```

Fixed Point Notations

Program

```
#include <iomanip>
const double pi = 3.14159;
double x = 2;
int main()
{
        cout << setprecision(3) << fixed;
        cout << setw(6) << pi << endl;
        cout << setw(6) << x << endl;
        // The setw() here is optional, but useful
        // Fixed changes the function of setprecision()
        // setprecision now sets number of decimal places
        cout << defaultfloat;   // remove the fixed option
        return 0;
}
```

Console

- `./a.exe`

```
3.142
2.000
```

Left and Right Justification

Program

```
#include <iomanip>
string str0 = “Hello World”;
int main()
{
        cout << setw(15) << str0 << endl;
        cout << left << setw(15) << str0 << endl;

        // Left justified continues until right is used.
        // Right justified is default.
        cout << right << setw (15) << “end\n”;
        return 0;
}
```

Console

```
➢ ./a.exe
     Hello World
Hello World
              end
```