C++ I/O Manipulators*

(CS 1002)

Dr. Muhammad Aleem,

Department of Computer Science,

National University of Computer & Emerging Sciences,
Islamabad Campus

*Acknowledgements: Prof. Tim Colburn, University of Minnesota-Duluth

Manipulators

A **manipulator** is a **simple function** that can be included in an **insertion** or **extraction chain**

- **C++ manipulators**
  - **must include iomanip** to use

Output Manipulators (no args)

Manipulators included like arguments:

- **endl** - **outputs** a **new line character**, **flushes output**
- **dec** - **sets int** output to **decimal**
- **hex** - **sets int** output to **hexadecimal**
- **oct** - **sets int** output to **octal**

Example:

```
#include <iostream>
#include <iomanip>
int x = 42;
cout << oct << x << endl; // Outputs 52\n
cout << hex << x << endl; // Outputs 2a\n
cout << dec << x << endl; // Outputs 42\n
```

Output Manipulators (1 arg)

Manipulators with 1 argument

- **setw(*int*)** - **sets** the **width** to ***int*** value
- **setfill(*char*)** - **sets fill char** to ***char* value**
- **left – Left Justified text**
- **setprecision(*int*)** - **sets precision** to ***int* value**
- **setbase(*int*)** - **sets int** output to **hex** if *int* is 16, oct if *int* is 8, dec if *int* is 0 or 10

```
cout << setw(7) << setprecision(2) << setfill(‘_’) << 34.267 <<
   endl;
  // outputs __34.27
```

Floating Point Format

- Can use flags **scientific** and **fixed** to **force floating point output** in **scientific** or **fixed format**

- Effect of precision depends on format
  - *scientific (total significant digits)*
  - *fixed (how many digits after decimal point)*

```
float a = 4.0;
int n=7;
cout<<"\nDefault:"<<setprecision(n)<<a<<endl;
cout<<"\nFixed:"<<setprecision(n)<<fixed<<a<<endl;
cout<<"\nScientific:"<<setprecision(n)<<scientific<<a<<endl;
```

Input Status Flags

- When **performing input**, certain **problems may occur**, we can **determine** if an **error has occurred** or **using these** by checking these **flags**:

  eof( ) - *end-of-file occurred during input*

  fail( ) - *input operation failed*

  good( ) - *no flags set (not eof or any of fail flags)*

- Flags stay set and all input fails until **clear( )** function called

Input Status Flags

- Examples… (Demo)

Character Input

- The **extraction operator** when applied to a **character** **ignores** **whitespace** (***space***, ***tab***, ***enter*** **key**)

- To **read any character** (i.e., ***space***, ***tab***, ***enter*** **key**) use the **get(*char*)** **function** with ***cin*** object, can also provide **no argument** (***works like getchar***)

```
char ch;
cin >> ch;     // Reads next non-whitespace char
cin.get(ch);   // Reads next character (any:
              //  space, newlines, tab, etc.)
```

String Input

- With **cin>>**, **whitespace** (***space***, ***tab***, ***enter*** **key**) **characters are ignored**

- To **get an entire line** (including **whitespace characters** such as **space** and **tab**) from **cin**,
  - Use **getline ( ) function**
  - Syntax: **getline (cin, str_variable);**

    getline (cin, str_variable, char termChar);

  - Example program….

Any Questions!