cout

- C++ uses a convenient abstraction called *streams* to perform input and output operations to or from sequential “media”, such as the screen, the keyboard, or a file. A stream is an entity that a program can either insert or extract characters to or from.
- C++ uses a much simpler method than C’s `printf()` called `cout`.
- You simply pass what you want to print as a stream into `cout`.

```
     cout << "Hello " << endl << " C++" << endl;
```

- `endl` for `cout` is essentially the same as the escape character `"\n"`

setprecision()

- Use `setprecision()` to set precision to display floating-point values using `cout`.
- When using `setprecision()` you will need to `#include <iomanip>`
- `cout << fixed;` Write floating-point values in fixed-point notation.
- `cout << showpoint;` Decimal point is always written for floating point values inserted into the stream, even for those whose decimal part is zero.
- `cout << setprecision(2);` As many digits as necessary are written to match the precision set for the stream (if any), e.g.,

```
     const double PI = 3.14159;
     cout << fixed; cout << showpoint;
     cout << setprecision(4) << PI << endl;
```

setw()

- Another useful function in `<iomanip>` is `setw()`.
- `setw()` can be used to print an arbitrary number of spaces into `cout`.

```
     cout << setw(10); cout << 77 << endl;
```

- `setw` needs to be declared immediately **before** the variable to be printed, declaring the total number of spaces to be used including the variable.
- There are also, `left` and `right` alignment declarations, e.g.,

```
   cout << fixed << showpoint << setprecision(2) << left;
```

cin

- The standard input stream is a source of characters determined by the environment. It is generally assumed to be input from an external source, such as the keyboard, or a file.
- Here is C++ code to add two `int` values together,

```
     int value1, value2;
     cout << "Enter 1st value: ";
     cin >> value1;
     cout << "Enter 2nd value: ";
     cin >> value2;
     cout << value1 << " + " << value2 << " = " << (value1 + value2);
```

- Try changing `int` to `char` in the above and see what happens.

getline()

- If you want to get a whole line of `string` input, rather than the first word use getline() instead,

```
     string str;
     getline(cin, str);
     cout << str << endl;
```