Programming Notes 2: Basic IO

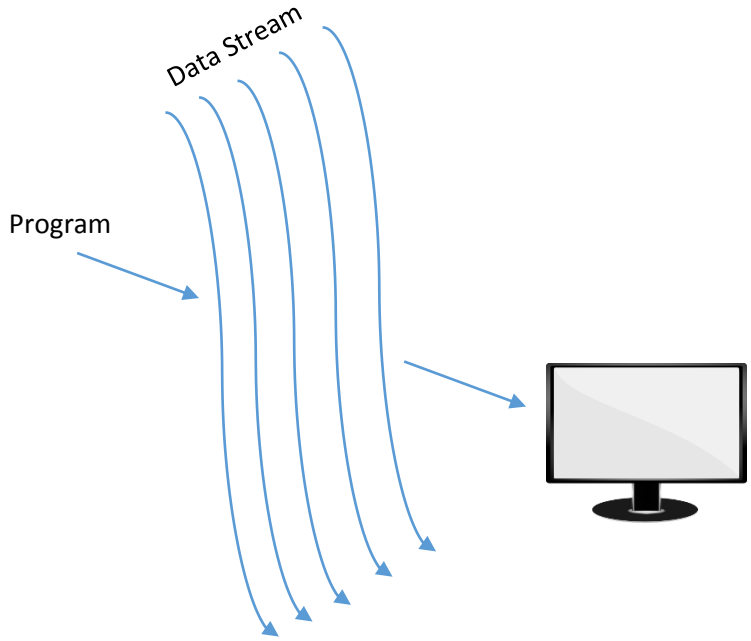
Always remember, I will provide examples from the notes in the example program.

1. Intro to IO

Now that you've had a quick introduction to data and variables, let's quickly go over how the basic input and output of your programs will work, so you will be able to experiment with the next few things you learn. Console input and output is very simple to use in C++, and we'll be using it for quite a while.

2. Data Streams

A "data stream" metaphor is used to both manage and understand the flow of data into and out of a program. The data stream is considered to be one path within the computer to which various devices can be connected. You can think of a data stream like, well, a stream: data flows along this data stream, and your program can put data into it, and it will end up somewhere else, or your program can take data from the stream that came from somewhere else.



The two devices you will use at first are considered to be the default I/O or input and output devices. That is to say, they are the ones to be used if no other options are chosen. You will use the keyboard, which is considered to be the "Console In" device, and you will use the monitor, which is considered to be the "Console Out" device. These terms are actually integrated into the code: the keyboard input device will be represented by a programming object called cin, and the monitor output device will be represented by a programming object called cout.

3. Operators

The input and output process also uses operators to indicate the flow of the data. Using these operators assumes that your metaphorical data stream is on the left side of your program code. The insertion operator, used for output from the program, will point to the left side of your screen as if it is inserting data into the data stream, and the extraction operator, used for input to the program, will point inward toward your program and data as if it is extracting data from the data stream. Again, we will learn how to use these in more complex ways later.

4. Using cout and cin

First of all, to use these data streams, you will need to include these two lines of code at the top of your program, as explained in the hello world program:

```
#include <iostream>

using namespace std;
```

To use console output, you will use the data stream with the identifier "cout," or, also unsurprisingly, console out. As mentioned previously, you can then write the insertion operator, <<, followed by the data you want to send to the console output. You can then add another insertion operator followed by more data, and so on. Finally, you can also send "endl" to the console, which will simulate pressing enter, ending the current line and creating a new one. Examples:

```
cout << "Hello World" << endl;

cout << "This demonstrates" << " sending text to the console.";
```

Now, to use cin, it is mostly the same as cout, except you use the extraction operator, >>. On the right of your extraction operator, you would have the variable you want to input into. Examples:

```
cin >> userAge;

cin >> character;
```

5. Special Characters

Quick note—if you want to include a character like a double quote in a string, which would normally end the string, first type a backslash, and then the character. For example,

```
"The person said \"hello\" to me"
```

This tells the compiler to include that character in the string, and the slash won't show up. You also have to use this for backslashes themselves (i.e. "\\" to output one slash).

6. Holding the screen for the user.

As mentioned with the hello world program, if you run your program with no way to hold the screen open, it will open and immediately close, not allowing the user to see your output. To display the message "Press any key to continue…" and wait for the user to press a key before continuing, use the line

```
system("pause");
```