

STANDARD INPUT AND OUTPUT*

Kenneth Leroy Busbee

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 2.0[†]

Abstract

An explanation of standard input and output as used in the computer industry followed by examples of cin and cout as used within a C++ program.

1 General Discussion

Every task we have the computer do happens inside the central processing unit (CPU) and the associated memory. Once our program is loaded into memory and the operating system directs the CPU to start executing our programming statements the computer looks like this:

*Version 1.5: Jan 13, 2010 3:38 pm -0600

[†]<http://creativecommons.org/licenses/by/2.0/>

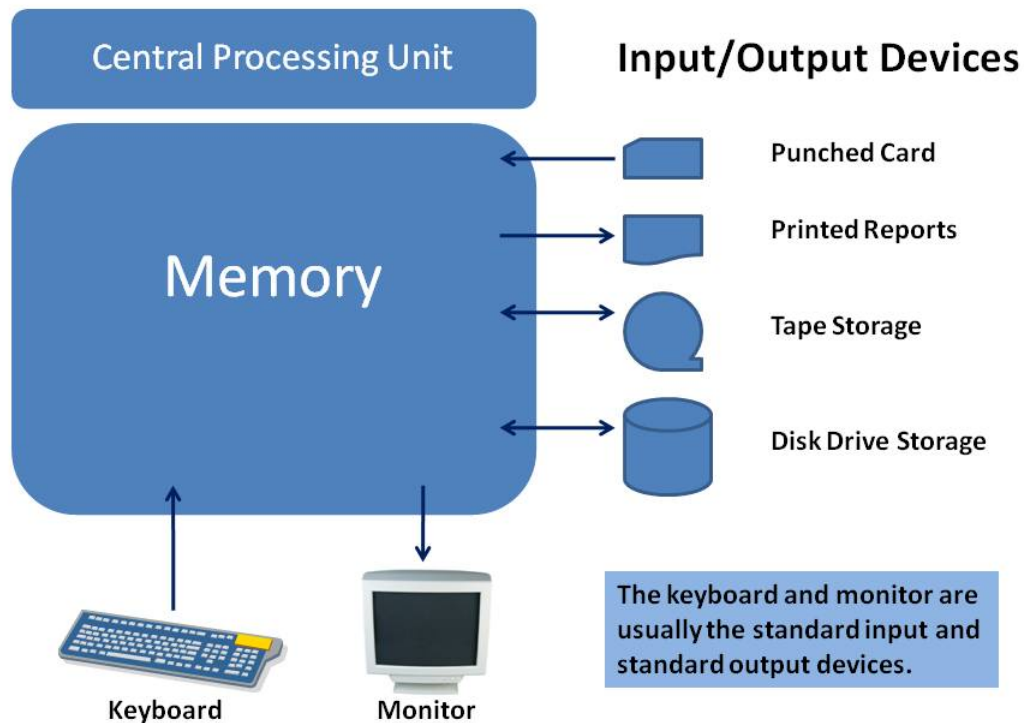


Figure 1: CPU – Memory – Input/Output Devices

Our program now located in the memory has basically two areas:

- Machine instructions – our instructions for what we want done
- Data storage – our variables that we using in our program

Often our program contains instructions to interact with the input/output devices. We need to move data into (read) and/or out of (write) the memory data area. A **device** is a piece of equipment that is electronically connected to the memory so that data can be transferred between the memory and the device. Historically this was done with punched cards and printouts. Tape drives were used for electronic storage. With time we migrated to using disk drives for storage with keyboards and monitors (with monitor output called soft copy) replacing punch cards and printouts (called hard copy).

Most computer operating systems and by extension programming languages have identified the keyboard as the **standard input device** and the monitor as the **standard output device**. Often the keyboard and monitor are treated as the default device when no other specific device is indicated.

2 Standard I/O within C++

The developers of the C++ programming language decided to provide some of the more technical code needed to interact with the operating system and the I/O devices. In the following example the include directive inserts a file that contains code from the **Input-Output Stream** library. This file contains necessary code to use **cout** and **cin** for sending data to the monitor or getting data from the keyboard.

```
#include <iostream>
```

You should think of **cout** and **cin** as being locations that you can send to or receive data from; similar in concept to any other variable storage location within the data area of our program. The C++ programming language has two operators to use in conjunction with I/O devices.

Action	C ++ operator symbol	Used with
insertion operator (write)	« (a pair of less than symbols)	cout
extraction operator (read)	» (a pair of greater than symbols)	cin

Table 1

Consider the following code:

Example 1: Insertion and Extraction

```
int age1;    // variable set up
             then later on in our program
cout « "\nEnter the age of the first person --->: ";
cin » age1;
```

Using the **cout** the programmer displays (or inserts) a prompting message on the monitor for the user to see. Using the **cin** the user types an integer value and hits the enter key and the computer extracts the value from the keyboard and stores it into the variable named `age1`. Within the computer all data are stored as numbers and thus part of the technical code provided by the developers of the C++ programming language that is within the **Input-Output Stream** library converts data from numbers to those symbols we are used to seeing as humans and vice versa. Example: If the user entered the numeral digits 57 and hit the enter key – the extraction operator would convert the 57 into a binary number and move the binary number into the integer storage place named `age1`.

The **cout** which uses the standard output device does not format the output into a Graphical User Interface (GUI) where you have a mouse to use. A modern operating system using GUI normally opens a **black screen output box** that would be similar to how the monitor was used when first developed in the 1960's. That is the default of how **cout** is normally implemented by most compilers.

The output message has a unique item worth mentioning. At the very front of the message is a backslash followed by the letter n. They do not get printed on the monitor. It is a special code (called a printer **escape code**) telling the printer to go to a new line. Printer! I thought we were using a monitor? We are but the code is a left over from the early days of printer output. The backslash tells the printer or monitor that the next letter is a command. The letter n is used for telling the printer or monitor to go to the front of a new line.

3 Definitions

Definition 1: device

A piece of equipment that is electronically connected to the memory so that data can be transferred between the memory and the device.

Definition 2: standard input

The keyboard.

Definition 3: standard output

The monitor.

Definition 4: insertion

Aka writing or sending data to an output device.

Definition 5: extraction

Aka reading or getting data from an input device.

Definition 6: escape code

A code directing an output device to do something.