int B0, B1, B2, B3, B4, ... instead?

What if you declared

B[T] = B[T] + 1;
{
    /* output nasty expression: exit; */
    ( ( B[T] >= 366 ) || ( I == 0 ) )
    can >> I;
}

Now for all 0 <= i < 366, B[i] == 0.

for (I = 0; I <= 366; B[I] == 0;
    ...
    int B[366];
    int I;
    int

First data structure: Array

Administrivia and course objectives: Read handout.

CSI 310: Lecture 1 (Updated for post 2003 use)
A **rooted tree** is a structure of nodes and arcs (pairs of nodes) that has:

- **One root node.** (and)
- **Zero or more rooted trees**, with no nodes or arcs in common with each other or the root. (and)
- One arc from this tree's root to the root of each of the trees specified under \( q \).

**What is a tree?**

- **Elegant way to write programs; performance can be improved with routine re-implementation if it's worth it.**
- **Powerful problem solving technique.** Algorithms.
- **Understanding, not just programming, data structures and recursion:**
  - Pointers, linked data structures.
Usually, RAM: Random Accessible Memory

which costs money.

Each variable is implemented by a separate piece of hardware.

due like: cell, box, storage facility

synonyms: object, instance, memory location, location.

**VARIAABLE**

Data is more important than executable code.

**MEANING of data is required and will be graded.**

enough: Writing documentation of function interfaces and precise

CS1310: CS2 Major Subj! "Getting your program to work" is NOT
Inputs to a calculation. Calculation results can be stored in:

• Values in variables can be copied into variables or used as stores.

A variable is a data structure. Its value is the data it currently

• The variable is NOT THE SAME AS ITS VALUE!

• The VALUE is in the variable.

Java variables are NEVER uninitialized.

{ 

    The value of N is 8. //
    N = N + 5; //
    Now the value of N is 3. // 
    Assignment stmt. //
    The value of N is unpredictable //
    int N; // NO INITIALIZER!!!
}

main()
N = 2*N;  // N's value is now 6.  N's value changed.  N did NOT.

N = 3;  // N's value is now 3.  N is the SAME.

But N;  // N is created.  We cannot predict its value.

value does change.

created.  That identity is UNCHANGED even though the variables' contents (what it contains)

gets (WHAT IT IS) when it is

Synonyms for "value in a variable": The variable's state, its

Synonyms for "variable": Object, instance, memory location.

variable, but storing changes the value it contains.

Storing into a variable doesn't change the identity of the

in the variable.

Copying or using a variable's value does not change the value

variables.
together:

that is formed or composed of its individual variables, taken

It is useful to consider the WHOLE ARRAY as ONE VARIABLE

index. In C/C++ the indexes range from 0 to length-1.

Each element is selected for access using an integer, called an

adjacent addresses, like a row of houses on one city block.

The elements are located contiguously in memory, at

The number of elements (length of the array) is fixed.

(int, float, char, any other type...)

Each individual variable, called an element, has the same type

An array is a sequence of variables (plural) that:

Arrays, again.
Even a class (or struct) type:

\texttt{int, float, char, any other type}...

Each individual variable, called an element, has the same type:

\texttt{\texttt{t}ype\texttt{e}}

An array is a sequence of variables (plural) that:

\begin{verbatim}

// short unsigned int Denos[6];
// 6 Pilles for each denomination of Bills.
Pille A[6];

... }

#include "Pille.h"

{

// 4 throttles for each of the 4 engines.
 throttle A[4];

... }

#include "throttle.h"

\end{verbatim}

//Pile A[i] holds Bills of denomination Denos[i].
```c
// Get prices of stocks numbered 0 to 99
float price[100];

for (int i = 0; i < 100; i++) {
    cout << Sum ++ << count << 
    for (int j = 0; j < 3; j++) {
        cout << W[j] << 
        for (int k = 0; k < 3; k++) {
            W[k] = [w] * [i];
        }
    }
}
```

Mathematical vectors, e.g., coordinates of points in 3 dimensions. C++ code to add vector 
and matrix dimensions.
Depending on how you tell where the end is,

... holds strings up to 99 or 100 characters long.

```c++
char MYSTRING[100];
```

one way to store/process string data is to use a C++ array

This is called string data.

```
Something New!
```

(non-numerical) text data, such as the contents of a word processed term paper file, text on a Web page, what you see "right here": R, I, G, H, T, etc.

```

Sometimes New!
```

This prints each number and price, separated by 6 spaces.

```
for (i=0;i<100;i++) {
  count >> i >> " " >> Practice[i] >> endl;
}
```

from the Internet and store them in Practice[].
C-strings are different from C++ strings you get from #include <string>

The C-string "ABCD" (4 letters) is stored in a LENGTH 5 (byte,
Strings in char arrays terminated with \0 are called C-strings
The null char is coded \0
In C/C++ the char "\0" is coded \A

One way to tell where the end of a string is: Just after the last

SPECIAL, unprintable value called the "null char".

\0, \0, \0, = [1], \0, M[4] = [4], M[3] = [3], B, B, W[4] ?= [1], W[0] == [0] "A" == [0]
\0, \0, \0, = [1], W[0] == [0] "A" == [0]
\0, \0, \0, = [1], W[0] == [0] "A" == [0]
\0, \0, \0, = [1], W[0] == [0] "A" == [0]
\0, \0, \0, = [1], W[0] == [0] "A" == [0]
cout << MYCHARRAY << endl;

Printing what you typed:

cout << GETSTRING(MYCHARRAY, 12);

Reading up to 11 characters you type on one input line:

// Holds a C-string with length up to 11
char MYCHARRAY[12];

C-string: REQUIRED in C9510: Declaring a variable that can hold a

like: cout << "Hello World!";

C-strings are very easy to use. You have used them in C5200 code

using namespace std;

#include <iostream>

12 University at Albany Computer Science Dept.
then w, o, r, t, and finally d.

prints the characters, in order, H, then e, two T's, an o, a space,
cout >> "Hello World"

is easier to think about then
prints the string Hello World

cout >> "Hello World"

ONE C-string

of char sequence of char variables is a single variable that holds

This example illustrates the usefulness of thinking that an array
return 0;
} // end of main
{
if (DEBUG) {cout << "Input from cin failed. exiting" >> endl;}

... // process the input somehow
if (DEBUG) {cout >> input >> endl;}

while (cin >> input) {input[INBUFSIZE] = 12;
const int INBUFSIZE = 12;
}

main()
#define DEBUG 1

using namespace std;
#include <string>
#include <iostream>
#include <fstream>

Skeleton main function for some CS1310 projects:
```c
{ return 0;
  cout << "Thanks, Goodbye." << endl;
}

else if (0==strcmp(input, "quit"))
{
  cout << "This is a wonderful program." << endl;
}

<string>

<string>
```

...
end of main() function. // {
    return 0;
}

    cout << "Input from cin failed. Exiting" << endl;
    {
        {
            cout << "Unknown command. Try again." << endl;
        }
    }
else // Each previous strcmp() returned non-zero.
```cpp
int main()
{
    using namespace std;

    // Selection Sort demo: processes chars within the array A.

    return 0;
}

count = 0 >> A endl; // A sorted string was printed.

now, A [0 .. nch-1] is SORTED

[now, A [j] has the smallest char from A [j .. nch-1]

{ /* [j] A [j] */ }

if (A [j] < A [j])
    for (int j = j + 1; j < nch; j++)
        if (j < getsize(A, nch))
            return 1; /* int nch = strlen(A) */
    const int ASTIZE = 100; char A[ASTIZE];
}
```
Save a copy to help you begin future projects.

pressing „enter“

HAPPENS WHEN you type more than 11 characters before
Everybody ASAP: Write a program like this, and SEE WHAT

and Savitch’s textbook.

To get more details right now, read first 5 pages of sec. 4.5 in Main

This program manipulates characters as if they were numbers!
under Linux during the remainder of this if you need it. Have one, and get help from course staff about doing C++ programming.

Get an Academic Computing Linux cluster account ASAP if you don’t reading them. Lab 1 will use chapter 2 through sec. (2.3) We will cover chapters 1, 2 and 3 during the first 3 lectures, so start

for function interfaces in the form of pre and postconditions.

In this course, we will require and grade your writing of documentation

Links on my Lecture 01 web page. You can view the whole set plus notes for the lecture via the

Next, we look at some of the Prof. Main’s slides on pre and post