Core of Chapter 3 (begin).

Interactive Test Driver pattern (From Ch. 3)

C-strings AKA null-terminated char arrays

Arrays.

LAB DISCUSSION and TRAINING I are SEPARATE.

Office Hour Announcement, last minute UNIX help after class.

email about internships, etc.

NEW CSIB-L@listerv.albany.edu—see me if you have NOT gotten

CSI 310: Lecture 3 Spring 2006

University at Albany Computer Science Dept.
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 indices 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.
```c
// 4 throttles for each of the 4 engines.

throttle A[4];

... }

#include "throttle.h"

Even a class (or struct) typedef
(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:
```


```c
{
#include "Pile.h"

Pile A[6];
// 6 Piles for each denomination of Coins
short unsigned int Denos[6];
Denos[0] = 1; Denos[1] = 5;
// Pile A[i] holds Coins of denomination Denos[i].
```
{ ; [I] = [I] + [I] M; for (I = 0; I < 3; I++)
{ ; [I] M <= [I] C; for (I = 0; I < 3; I++)
{ ; [I] M <= [I] C; for (I = 0; I < 3; I++)}
}
}

int I;

count >>> TYPE vector M;


double V[3][3, S];

dimensions. C++ code to add vector A and M:

Mathematical vectors, e.g., coordinates of points in 3

What can you use arrays for?
This prints each number and price, separated by 6 spaces.

```cpp
{ count >> i >> pr1ce[i] >> endt;
    count >> i >> pr1ce[i] >> endt;
    }
for(t=0;t<100; t++)
    count >> stock number pr1ce >> endt;
    count >> stock number pr1ce >> endt;
// From the Internet and store them in pr1ce[].
    for(float pr1ce[100])
        // Get prices of stocks numbered 0 to 99
        for(float pr1ce[100])
        // Get prices of stocks numbered 0 to 99
    }

Statistics, such as prices of 100 different stocks:
Depending on how you tell where the end is.

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

char mystring[100];

of char, declared:

One way to store/process string data is to use a C/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, C, H, T, etc.

Something New! Non-numerical "text" data, such as the
C-strings are different from C++ strings you get from #include <string>

\10, \4 == [M] \2 == [B], \3 == [C], \1 == [A], \5 == [W]

not \4 (char [W]) with:
The C-string "ABCD" (4 letters) is stored in a LENGTH 5 (five,
Strings in char arrays terminated with \0, are called C-strings
The null character is coded \4,
In C/C++, the char "\4" is coded \A,

SPECIAl, unprintable value called the "null char".
One way to tell where the end of a string is: just after the last

University at Albany Computer Science Dept.
cout << MYPHRASE " endl;
Printing what you typed: cout

getLine (MYPHRASE, 12);
Reading up to 11 characters you type on one input line:

// Holds a C-string with length up to 11

char MYPHRASE[12];
C-string:

REQUERED in CS1310: Declaring a variable that can hold a

like: cout << "Hello World";
C-strings are very easy to use. You have used them in CS1201 code

using namespace std;

#include <iostream>
then W, o, x, 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 than

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

University at Albany Computer Science Dept.
```cpp
return 0;
}
#endif

int main()
{
    cout << "Input from cin failey. exiting" << endl;
    ...
}
```

```cpp
while(cin.getline(input, INBUFSIZE))
    char input[INBUFSIZE];
    const int inBUFSize = 12;
}
```

```cpp
main()
#define DEBUG 1
using namespace std;
#include <string>
#include <iostream>
#include <iostream>
```

Skeleton main function for some CS1310 projects:
```c
{ 
  return 0;
  cout << "Thank you!" << endl;
}

if (0 == strcmp(prompt, "quit")
{
  cout << "Hello!" << endl;
}

if (0 == strcmp(prompt, "about")
{
  cout << "About the program." << endl;
}

if (0 == strcmp(prompt, "help")
{
  cout << "Help commands:

  Recognizing one-line commands:
```
// end of main() function.

return 0;

cout << "Input from cin failed. exiting" << endl;

cout << "Error unknown command, try again." << endl;

cout << "Each previous stripped() returned non-zero."

else // Each previous stripped() returned non-zero.
So, their design choices were rational. But they did think that every last microsecond of computer time was valuable. Imagine that some nasty people will make that happen on purpose. The earliest C/Unix/Internet/DOS/Windows designs did not so-calle... BUFFER OVERFLOW!

When you copy a C-string without counting the characters so that

VIRUS AND OTHER SOFTWARE EXPLOITATION, VULNERABILITIES THAT ENABLES PEOPLE TO WRITE C-STINGS ARE PROBABLY THE LEADING

Cee, that's wonderful. Wow!
What are two critical issues for array?
viruses.

Others, worse, will produce hidden errors or openings to
otherwise, some systems will make your program CRASH.
that is too small or too big.
ensure your program never uses an element whose subscript
2. You must know where the ENDS OF THE ARRAY ARE, and
element one after the end.

(b) Put a special value (like the NUL, which = 0 character) in the
of elements.

(a) Have a constant or variable NUMBER that signifies the number

Two ways to know where the end is:

might vary...

The portion of the array allocation that is ACTUALLY USED
{ /* [j] } to code to "swap" A[i],A[j] */

if(A[j] < A[i]) {
    for(int j = i+1; j < nch; j++)
        temp = A[j];
    for(int j = i; j < nch-1; j++)
        A[j] = A[j+1];
    A[i] = temp;
}

return 1;
}

const int ASIZE = 100;
char A[ASIZE];

main()
{
    using namespace std;

    #include <iostream>

    Selection sort demo: processes chars within the array A.
Save a copy to help you begin future projects.

Pressing "enter" you type more than 11 characters before
HAPPENS WHEN
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 chars as if they were numbers!