Finned Array Data Structure.

5. Sequence-type container classes implemented with the partially

handouts.

Read ALL of ch1, ALL of ch2, begin ch3 for project, ALL of both

4. An old throtltee, intro to classes

3. assert() macro, proper and improper use


1. Prof. Main's colorful powerpoints on pre and post-conditions.

0. Project 1 Assessment is here, NOW.

CSI 310 Lecture 3
Useful invariants are about states/values of program variables, but

must be TRUE.

An invariant is correct, or true, or is not a lie means:

An invariant must have a PARTICULAR SPOT in a computer

and a predicate (e.g., “is now 3”).

A grammaratical sentence. It must have a subject (e.g., “N’s value”) a

An invariant is kind of expression of a thought (in your mind) by a

An invariant is NOT a kind of programming language statement. A

called invariants

sentences like “N’s value is now 3”, “and N’s value is now 6”, are
invariants.
form of useful pre and postconditions about the function.
interfaces by writing comments into the header file that are in the
One process requirement in this course is to DOCUMENT

What?
Study at the CONTACT on page 7. Who can sue whom for
postconditions.

preconditions and
interfaces by writing kinds of invariants called
Main and Savage chapter 1 covers how to document function

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/++, the indexes range from 0 to Tenth-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.
float price[100];

Statistics, such as pieces of 100 different stocks:

```
for(t=0; t<3; t++)
    for(j=0; j<2n; j++)
        sum[t] += [t][j];

int i;
    count >> Type vector W:
        for(t=0; t<3; t++)
            for(j=0; j<[t][j]; j++)
                sum[w][j] += [t][j];

    double v[3][w][3], sum[w];
```

What can you use arrays for?
Depending on how you tell where the end is.

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

```cpp
char MyString[100];
```
of char, declared:

This is called `string` data.

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

(Something New?): Non-numeric, "text" data, such as the

```
contents of a word processed term paper file, text on a Web
```

```
page, what you see "RIGHT HEREx: R, I, C, H, T, etc.
```

```
•
```

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

```
|
```

```cpp
for(t=0; t<100; t++)
```

```cpp
count << t << " Practice[]
```

```
}/from the Internet and store them in Practice[].
```

University at Albany Computer Science Dept.
C-strings are different from C++ strings you get from #include "string". The C-string "ABCD" (4 letters) is stored in a LENGTH 5 array:


not A (char W[0] with:
The null char is coded 0, the char "A" is coded A.

In C/C++ the null char is used for the string "A", the next element's value is the very special, unprintable value called the "null char". One way to tell where the end of a string is: Just after the last
cout << Mychartray << endl;

Printing what you typed:

cin >> MyChartray; //
Reading up to 11 characters you type on one input line:

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

REQuired in CS1310: Declaring a variable that can hold a
C-string

like: cout << "Hello World";
C-strings are very easy to use. You have used them in CS1301 code
using namespace std;

#include <iostream>
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 than

prints the string "Hello World"

  cout << "Hello World"

\begin{OneCStarIn}
\text{of char (sequence of char variables) is a single variable that holds}
\end{OneCStarIn}

This example illustrates the usefulness of thinking that an array
So, their design choices were rational. TheWas valuable.

But they did think that every last microsecond of computer time

Imagine that some nasty people will make that happen on purpose.

The earliest C/Unix/Internet/DOS/Windows designs did not

so-called BUFFER OVERFLOW!

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

VIRUS and other SOFTWARE EXPLOITORS.

VULNERABILITIES that enables people to write

C-strings are probably THE LEADING

Gee, that's wonderful. Wow!
```cpp
{ return 0; }
while (finished || getline(failed || empty string was read) == 0) {
    cout >> A >> endl; // A sorted string was printed.
    if (now, A[0..nch-1] is SORTED) {
        now, A[i] has the smallest char from A[i..nch-1]
        {
            [i..A[i]], A[i]
            ( [] < A[i] 
                for (j = i+1; j > nch; j++)
                    for (t=0; t < nch-1; t++)
                        while (cin.getline(A, A.size) && A[0..i] == 0)
                            const int A.size = 100; int nch, i; char A[A.size];
     main()
     using namespace std;
     #include <iostream>
     #include <iostream>

     // Selection sort demo: Processes chars within the array A.
}
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
Savitch's textbook.

To get more details right now, read pages 183-187 of Main and

This program manipulates chars as if they were numbers!
True on the call throttle(5)
False on the calls throttle(-1), throttle(0)

Precondition: size > 0.

Type of that parameter is integer.

"int size" specifies it has exactly one function parameter and the throttle( int size) is a prototype

True? False? A jury can determine that

and the object.

is a complete English sentence ("is" is the verb, note the subject

"Proct. Chicken is a turkey"

pre' post-conditions are definite statements about process state:

C/++ (standard (assert ( macro

You, so you find the bugs before your customers do!

You can easily command the computer to check your invariants for
Judged true or false.

sentences about values and states of objects, and can be denotely

Note the grammatical form of these statements: They are all full

constructor function.

size refers to the parameter value for the call to the throttle

shut off position, AND it IS currently shut off.

Postcondition: The throttle has size positions above the
(of course, there can be bugs in both)

function, or if it occurs in the code that calls the function, to distinguish whether a bug occurs inside the implementation of the assertion to check a precondition can enable the programmers

3) if value \( i = 0 \), go on.

number the assertion was coded.

here’s (with a nice message that tells in which source file and line

2) if the value \( = 0 \), make the process CRASH, (the assertion

1) evaluates the expression.

assert \( useless \) expression;

body an assertion to check the function’s precondition.

Highly Recommended: code at the beginning of each function
do anything needed for program function. For the same reason, expressions evaluated by assert() must not

Important:

done when performance (running speed of the software) is

why assert() expressions are simply omitted when the program

Check for user input errors, or resource limit failures.

Improper use of assert():
value of top-position.

Avoid making the value of position become negative, or to exceed the positive non-zero integer. Finally, to shift the throttle, make code that

like position = 0. Also, remember to assign top-position to some

kindly, please be nice and follow these tricky rules. But do anything

Dear Programmer,

probably won't bother; something like this:

Therefore, you might write comments or other documentation (but you


int position;

C/++ code you write is

An old-fashioned style for implementing a "throttle": The ONLY
nothing protects them from errant access.

Disadvantages: position and top-position are global variables:

```java
{ 
  position = 0;
  (0 > position) &&
  top-position = position
} 

void shutdown()

{ 
  position = 0;
  x = position
} 

void initialize()

variables by calling functions like these:
and warn everybody that they should only access the two throttle
and small improvement is to code "throttle maintenance functions"
It is very clumsy if you want to upgrade your software to have multiple throttles.

Functions used for different program objects.
The names of the functions will clash with initialization and other
be negative!

that puts the throttle in an illegal state. The position should never

Notice nothing prevents an errant programmer from writing code

throttle; throttle.position = -38;

written like:

code that defines a separate throttle and then accesses it can be

;

} throttle

top

} throttle

struct throttle

In a header file, declare the throttle type as a structure:

Here's a way to overcome the 3rd disadvantage:
class Throttle

In the throttle.h header file, declare the throttle type:

Here's the object-oriented way:

{?

  throttle();

  void throttle(int x);

  public:
  int top;
  int position;

  private:
  }
```cpp
{ 
  position = 0;
  top = 1;
}
__throttle::throttle()
{

  position = 0;
  (position > 0) {
    position = top;
    (top < position) {
      position = position + x;
    }

    (void __throttle::shift(x)
      begin thunks: }

    In the __throttle::cxx implementation file, you, as the implementer of the __throttle, carefully code private member functions to do the
```
class
outside the body of a function member belonging to the throttle
REUSE to compile a private member access like mytr = top = 9;
The private member protection rules of C++ make the compiler
....

mytr = shift(1); // Now mytr is ON.
because the default constructor was called.
mytr is a properly initialized throttlethroat;
....

#include "throttle.h"

Files whose code defines and uses a throttle can then be written:
that the programmer coded the throttle class had designed.

C++ and Java ENFORCE the rules for working with a throttle

(similar features)

all functions to access and manipulate it) in one place. (Java has

encapsulate everything (variables to store the throttle's state plus

Object Oriented Programming features of C++ to

What we surveyed, and DSo details in Chap. 2, is how to use
Technology for Speed

- Clever data structures and algorithms

Abstract Data Types

CS2 "Data Structures" subject:

- Using arrays.

They and students must know inners first

C++ professionals use Standard Template Library

Directly useful in application programs

• Intro. to container classes

Chapter 3 of DS0:

CS1 310: Lecture 3
SION and special cases.
other calculations AVOID DIVI-
WITH homogeneous coords, many
\( m/y = \lambda \) and \( m/x = \lambda \). 
pute \( X \) and \( m/x = \lambda \). 
tive coordinates. To plot, com-
used as homogeneous or projec-
Triple of points \( (x', y', \lambda) \).

ALTERNATIVE:
Pair or double host nums.
double(pos)/double(max)

DIVIDE
Algorithm: To calculate how,
pos and max
Pair of integer variables.

IMPLEMENTATION DIvE
Abstract Data Type

```c
{...
GetX(); GetY();
rotate90();
shift();
}
```

throttle();

```c
constructor
```

flow();
shift();

throttle

```c
```
structure/algorithm implementation examples.
abstract data types and for clever, efficient data helpful. CS1310 will concentrate on container classes both as
throttle and point are examples of concrete classes; very
... 0 1 2 3

How many times does \( x \) appear in \( M \)?

Given an item \( x \),

\textbf{Multiset} \( M \) (another name for "bag")

\textbf{No}

\textbf{Yes}

\textbf{No}

has a well-defined answer:

"Is \( x \) in \( S \)?"

Given an item \( x \),

\textbf{Set} \( S \)

\textbf{Yes}

What is a bag? Mathematicians say "finite multiset"
<table>
<thead>
<tr>
<th>Item Name</th>
<th>No. orders</th>
<th>Item number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grilled cheese sandwich</td>
<td>1</td>
<td>99</td>
</tr>
<tr>
<td>Pot of hot &amp; sour soup</td>
<td>3</td>
<td>27</td>
</tr>
<tr>
<td>Egg roll</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>Szechuan chicken</td>
<td>5</td>
<td>72</td>
</tr>
</tbody>
</table>

Example: What a Chinese restaurant serves:
implementations. They often vary in efficiency.

The same abstract data type can have very different implementations.

```plaintext
of bag

mentation for various variants

ALTERNATIVE

The linked-list (to be taught) is

Partially filled array.

\[
\begin{array}{ll}
\text{keyed_bag} & \{ \ldots \} \\
\text{exercises} & \{ \ldots \} \\
\text{see ch. 3} & \{ \ldots \}
\end{array}
\]

bag-with-receipts

Implementation Data Structure

Abstract Data Type

More abstract data type/implementations.