

The slide features a light blue background with abstract circuit-like patterns in purple and orange. These patterns include lines, dots, and geometric shapes that resemble electronic components and wiring, scattered across the top, bottom, and right sides of the slide.

Pointers, Arrays, and References

CSE 232 – Dr. Josh Nahum

Reading:

Section 1.7

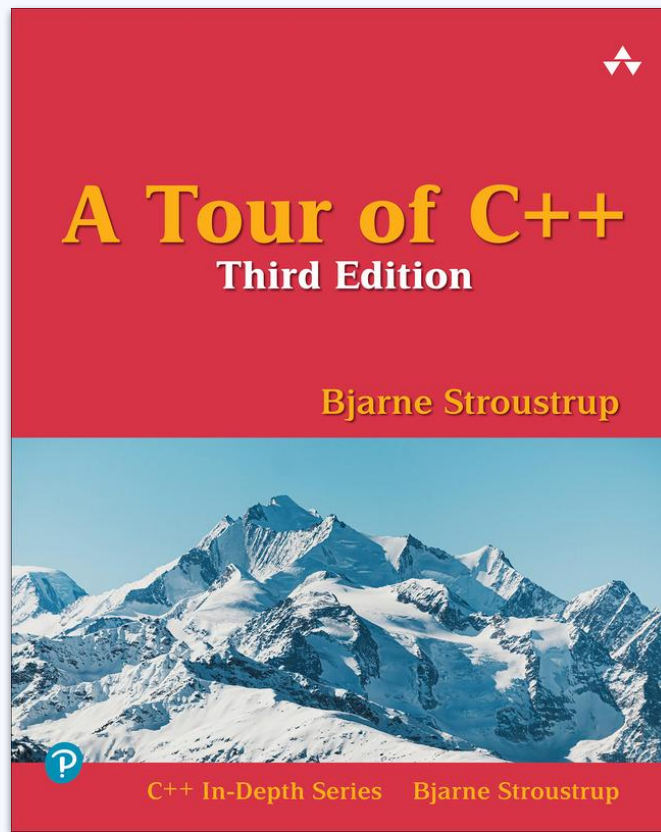




Table of contents

00

`&, *, []`

01

`reference to const`

02

`const and pointers`

03

`C-style strings`





00

&, *, []



Classification

| | * | & | [] |
|--------------------------|---|--|---|
| Declaration | <ul style="list-style-type: none">• "pointer to"• <code>int * x;</code> <p>x is a pointer to an int</p> | <ul style="list-style-type: none">• "reference to"• <code>int & x;</code> <p>x is a reference to an int</p> | <ul style="list-style-type: none">• "array of"• <code>int x[y];</code> <p>x is an array of ints of length y</p> |
| Unary Expression | <ul style="list-style-type: none">• "contents of" / "dereference"• <code>*x;</code> <p>dereference the pointer named x</p> | <ul style="list-style-type: none">• "address of"• <code>& x;</code> <p>the address of x</p> | |
| Binary Expression | <ul style="list-style-type: none">• "multiply"• <code>x * y;</code> <p>multiply x with y</p> | <ul style="list-style-type: none">• "bitwise and"• <code>x & y;</code> <p>bitwise and x with y (not relevant for 232)</p> | <ul style="list-style-type: none">• "element of"• <code>x[3];</code> <p>the fourth(!) element of x; zero indexed</p> |



01

reference to const



reference to const

reference to const

Example:

```
int const & x = y;
```

x is a reference to y, which is treated as a constant int

Note: y need not be const, and the value of y can't be changed through x

reference to non-const

Example:

```
int & x = y;
```

x is a reference to y, which is treated as a non-const int

Note: y must be non-const



02

const and pointers

pointer to const

Like references, you can have a pointer to const.

Example: `int const * x`

x is a pointer to a constant int

You can't change the value that x points at.

const pointer

To make a pointer that can't be changed to point at a different object, you need a const pointer.

Example:

```
int * const x
```

Note that the const is on the right side of the *. A const pointer to a non-const int.

```
int y = 78;  
int * const x = &y;  
*x = 12; // the value of y is now 12  
int z = 45;  
x = &z; // Compiler Error!  
// x is a const pointer  
// It can't point elsewhere
```



03

C-style strings



C-style strings

Declaration

```
char x[10];
```

Declares an array of
10 characters

Value

```
{'h', 'e', 'l', 'l', 'o', '\0'}
```

or
"hello"

String Literal

"hello" is a string literal and
is immutable (the characters
in it can't be changed)

Null Character

'/' is the null character, it has
the value 0
'/' == 0; // true
All C-style strings must end with
the null character



Attribution

Please ask questions via Piazza

Dr. Joshua Nahum

www.nahum.us

EB 3504



CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)

© Michigan State University - CSE 232 - Introduction to Programming II