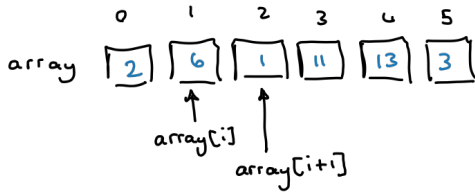


Today: Principle of least privilege ; how much memory with sizeof

Computer Programming 143 – Lecture 19

Pointers III



Faculty of Engineering



7.5 Using the const Qualifier with Pointers I

const qualifier

- Variable cannot be changed

```
const double PI = 3.141592653589793238;
```

- Use const if function does not need to change a variable
- Attempting to change a const variable produces a compiler error

7.5 Using the const Qualifier with Pointers II

const qualifier with pointers

There are 4 possible combinations:

- 1 Non-constant pointer to non-constant data

```
int* myPtr;
```

- Both the value (address) of myPtr and the integer it points to (*myPtr) may be changed

- 2 Non-constant pointer to constant data

```
const int* myPtr;
```

- The value (address) of myPtr may be changed, but the integer it points to (*myPtr) may not

7.5 Using the const Qualifier with Pointers III

const qualifier with pointers

8 Constant pointer to non-constant data

```
int* const myPtr = &x;
```

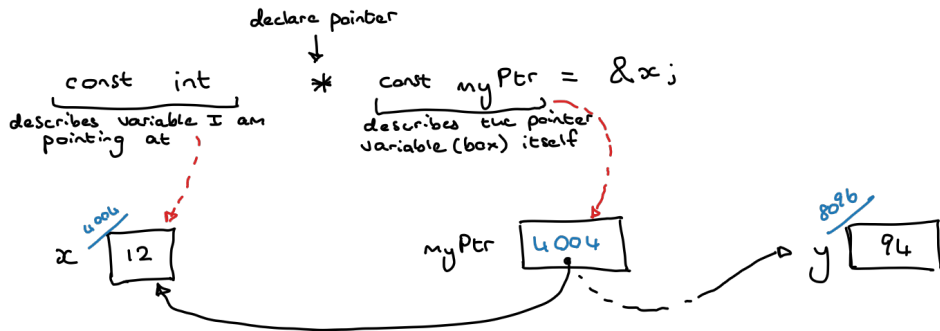
- The value (address) of myPtr may not be changed, but the integer it points to (*myPtr) may – myPtr points to an unchanging memory position
- The value (address) of myPtr must be initialised at declaration

9 Constant pointer to constant data

```
const int* const myPtr = &x;
```

- Neither the value (address) of myPtr nor the integer it points to (*myPtr) may be changed
- The value (address) of myPtr must be initialised at declaration

const with pointers



Why use least privilege?

- Protect yourself from others screwing up
- " " " yourself " "

7.5 Using the const Qualifier with Pointers IV

When to use which combination?

- Applicable to argument declaration when writing function definitions
- Use the *principle of least privilege*
- Prevents errors – ensures that your function does not accidentally alter data

7.5 Using the const Qualifier with Pointers VIII

Example: non-constant pointer to constant data (Fig.7.7)

```
#include <stdio.h>

void printCharacters( const char* sPtr );

int main( void ) {
    // initialise char array
    char string[] = "print characters of a string";

    printf( "The string is:\n" );
    printCharacters( string );
    printf( "\n" );
    return 0; // indicates successful termination
} // end main
```

7.5 Using the const Qualifier with Pointers IX

...Example: non-constant pointer to constant data

```
/* sPtr cannot modify the characters to which it points,  
 * i.e., sPtr is a "read-only" pointer */  
void printCharacters( const char* sPtr ) {  
    // loop through entire string  
    while (*sPtr != '\0') { // no initialisation  
        printf( "%c", *sPtr );  
        sPtr++;  
    } // end for  
} // end function printCharacters
```

Output

The string is:
print characters of a string

“Sizes” of variables, i.e. the sizes of the boxes

- 1 bit = one 0 or 1
- 1 byte = 8 bits
- What is the size of one integer box?

