Pointers and Strings

Pointers and Functions

- What are the two ways of passing arguments into functions?
- Write two functions **`square1`** and **`square2`** that will calculate the square of an integer.
  - **`square1`** should accept the argument passed by value,
  - **`square2`** should accept the argument passed by reference.

Pointers and Functions

- There is a third way of passing arguments into functions
- It's called passing by reference without using reference arguments
- The address of the argument is passed instead of the argument itself

Passing by reference

```
void square3 (int *pNum)
{
  *pNum *= *pNum;
}
```

- What would a function call to the above function look like?

Function Call

```
intval = 5;
square3( &intval );
cout << intval << endl;
```


Strings

- What is a string in C++?
- How have we declared string variables? We have used two ways.

Strings

- In C++, strings are arrays of characters that end in the null character \0
- A string can be declared as:
  - `char pet[] = "cat";`
  - `char *pPet = "cat";`

Strings and Pointers

- When declaring and array, the name of the array is also a constant pointer to the first element in the array

```
int array[] = {2, 4, 6, 8};
int *pArray;

pArray = array;
pArray = &array[0];
cout << array[2]
     << *(pArray + 2);
pArray ++;
array ++; // ERROR
```


Strings

- Assuming that the string pet has been declared as:
  - `char pet[] = "cat";`
- Write a function that will output the contents of the string. The function should accept the array and its size
- Write a function that will output the contents of the strings. The function should accept a pointer to char

Strings and Pointers

- Write a function strLength that accepts a string (as a pointer) and returns the length of the string

Strings and Pointers

```
int strLength (const char *pStr)
{
  int index;
  for (index = 0; *(pStr + index) != '\0'; index ++);
  return index;
}
```

- What is the purpose of const in the function header?
- Is the ; at the end of the for loop a mistake?
- What would happen if the ; was eliminated?

Pointer Arithmetic

```
int strLength2 (char *pStr)
{
  char *pTemp = pStr;
  while (*pTemp)
    pTemp ++;
  return pTemp - pStr;
}
```

What is happening?

```
int sumInts (int *pArray, int size)
{
  int sum = 0;
  int index;

  for (index = 0; index < size; index ++)
    sum += * pArray ++;

  return sum;
}
```

- `int array[] = {10, 20, 30, 40, 50};` creates an array as follows:

Address	Value	Element
2000	10	0
2004	20	1
2008	30	2
2012	40	3
2016	50	4

Summary

- Today I introduced
  - The relationship between pointers and arrays
  - Pointers and strings
  - Pointer arithmetic
- We have covered:
  - P. 341 - 349
  - P. 360 - 362