Pointers and Strings

Strings

- A string can be declared as:
  - **`char pet[] = "cat";`**
  - **`char *pPet = "dog";`**
- In the second declaration above, **`pPet`** points to the first letter (**`d`**) of the string on the right
- Depending on the compiler, the space in memory that contains **`dog`** may or may not be modifiable
- If a string literal is to be modified, it should always be stored in a character array

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 string. 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?
- Why is the null character **`\0`** in single quotes?
- Is the **`;`** at the end of the for loop a mistake?
- What would happen if the **`;`** was eliminated?

Pointers and Strings

```
int strLength1 (const char *pStr)
{
  int index;
  for (index = 0; *(pStr++) != '\0'; index ++);
  return index;
}
```

- Will the above function give the same output as the function on the previous slide?

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

Constant Pointers

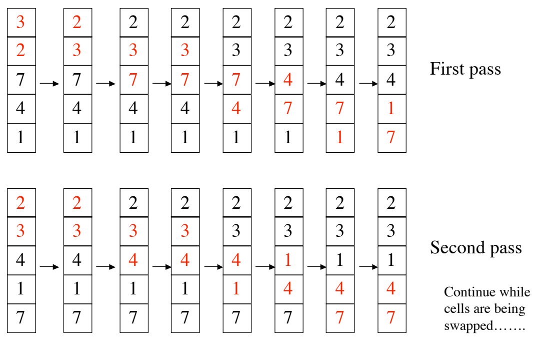
- So far we have seen:
  - Nonconstant pointers to nonconstant data
  - Nonconstant pointers to constant data
- What about constant pointers?
- We said that array names are constant pointers to the first element in the array. What does that mean?

Constant Pointers

```
int * const pNum, num, num2;
num = 9;
num2 = num + 8;
pNum = &num;
*pNum *= 2;
pNum = &num2;   // ERROR
```

- pNum has been declared as a constant pointer
- It cannot point to any other memory location

Bubble Sort



Bubble Sort

- Let's look through the code that will perform the bubble sort described on the previous slide
- http://zeus.cs.pacificu.edu/shereen/cs250/lectures/04bubble.html

Arrays of Pointers

- What do you make of the following declaration?

```
char *cardSuits[4] = {"Clubs", "Diamonds",
                      "Hearts", "Spades"};
```

- What gets output in each of the following cases?

```
cout << cardSuits[1] << endl;

cout << *cardSuits[1] << endl;
```

Summary

- Today I introduced
  - Pointers and strings
  - Constant pointes
  - Bubble sort
  - Arrays of pointers
- We have covered:
  - All of chapter 5