CPS235 Object Oriented Programming in C++

LAB7 BESE-15B *6th May 2010*

Using Pointers in C++

Objectives

By the end of this lab, you should be able to

- Declare and use pointer to basic data types
- Understand the relationship between pointers and arrays
- Use pointers as function arguments and return types
- Declare and use pointers to structures
- Using the **`const`** keyword with pointers

Question 1:

(a) Write a C++ program to input 3 integers. Set up a single pointer to point to each of these integers in turn. Display the values by dereferencing the pointer. The program's output would be like this:

```
Enter three integers: 24 30 55
The three integers are:
x = 24
y = 30
z = 55
```

(b) Modify the program in (a) to achieve the same results using an array of three pointers. Using the pointers in your array, rotate the values of the variables so that the value of x goes to y, that of y goes to z and that of z goes to x.

Question 2:

Write a C++ function that takes two integer arrays and their size (both arrays are the same size) and returns a 1 if the arrays have the same contents or a 0 if not. Use one pointer each to traverse the arrays for comparison. The interface of the function should look like:

```
int compare (int array1[], int array2[], int size);
```

Question 3: Pointers as Function Arguments and Return Types

For the following functions, **use the pointer notation (*) ONLY**. Do NOT use the array index [] notation

(a) Write a function **`countEven(const int* a, int size)`** which receives an integer array and its size, and returns the number of even numbers in the array.

(b) Write a function that returns a pointer to the maximum value of an array of double's. If the array is empty, return NULL.

```
double* maximum(const double* a, int size);
```

(c) Write a function **`myStrLen(const char*)`** which returns the length of the parameter C-string. Write the function without using the C++ function strlen.

(d) Write a function **`contains(const char*, char)`** which returns the pointer to the first (left-most) occurrence of the 2nd parameter char in the 1st parameter C-String, or NULL if the C-String doesn't contain the char.

```
const char* contains(const char*, char)
```

(e) Write a function **`revString(char*)`** which reverses the parameter C-String. The function returns nothing. You may use C++ string handling functions in <cstring> in the function if you wish. However, you cannot use the in-built string reverse function in the cstring library.

Question 4: Pointers with Structures

Recall that there is a minor difference between C++ structures and classes. So, you can use pointers to structures in the same way that you use pointers to objects.

(a) Declare a structure Phone that has three int fields called country, area and number. In main, declare a pointer variable to a Phone structure named newPhone. Then write assignment statements to indirectly (using the pointer) store the values 1, 888 and 5551122 into these fields.

(b) Write a Boolean-value returning function called ShallowCompare that takes two variables of type pointer to phone structure (declared in parta), and returns true if they point to the same structure, and false otherwise.

(c) Write a Boolean-value returning function called DeepCompare that takes two variables of type pointer to phone structure (declared in parta), and returns true if the structs that they point to have identical values in their corresponding fields.

Question 5: Using `const` with Pointers

Write a simple main function in which:

- Declare two integers
  - Give them different initial values
  - Make one of them const
- Declare four pointers, all initialized to 0
  - Make one of them a pointer to integer
  - Make one of them a const pointer to integer
  - Make one of them a pointer to const integer
  - Make one of them a const pointer to const integer
- Try assigning addresses of integer variables
  - Which pointers cannot change at all?
  - Which pointer can point to either variable?
  - Which pointer can only point to one of the variables?
- As you change the pointers
  - Print out their names, addresses, and values of what they point to