

C/C++ Programming

Southern University of Science and Technology

Mengxuan Wu

12212006

Midterm Review

Mengxuan Wu

1 Compile and Run

In C:

- Compile: `gcc -c hello.c`
- Link: `gcc hello.o -o hello`
- Run: `./hello`

In C++:

- Compile: `g++ -c hello.cpp`
- Link: `g++ -o hello hello.o`
- Run: `./hello`

“sizeof()” Function

When you use “sizeof()” function on an array, it will return the size of the whole array in bytes. For example, if we have `int A[2][3]`, then `sizeof(A)` will return 24, `sizeof(A[0])` will return 12, and `sizeof(A[0][0])` will return 4.

When you use “sizeof()” function on a string type variable, it will always return 32.

When you use “sizeof()” function on a pointer, it will return the size of the pointer itself, which is 8 bytes on a 64-bit machine.

Input a String

When input a string in C, we don’t write “&” before the string variable. For example, we write “`scanf("%s", str);`” instead of “`scanf("%s", &str);`”.

In C, to input a whole line, we use “`gets(str);`” instead of “`scanf("%s", str);`”. And we can use “`puts(str);`” to output a char array. Note: it is a must to add a null character “\0” at the end of the char array or will cause unexpected behavior.

In C++, there are two ways to input a whole line as a char array. We can use “`cin.get()`” to input a whole line, and use “`cout`” to output a string. “`cin.get()`” takes two parameters, the first one is the char array, and the second one is the maximum length of the char

array. However, if the input exceeds the maximum length, it will only read (maximum length - 1) characters and leave the last one as “\0”. We often use a no parameter version of “cin.get()” to read a linebreak character, so that the next input will not be skipped.

We can also use “cin.getline()” to input a whole line, and use “cout” to output a string. “cin.getline()” takes two parameters, the first one is the char array, and the second one is the maximum length of the char array. It behaves the same as “cin.get()” when the input exceeds the maximum length. But it behaves differently from “cin.get()” when countering a linebreak character. “cin.getline()” will read the linebreak character and discard it, so that the next input will not be skipped.

In C++, if we want to input a whole line as a string, we can use “getline(cin, str);”. It will read the whole line and discard the linebreak character, so that the next input will not be skipped.

Pointers

In C, we use “%p” to print the address of a variable. In C++, if you want to print the address of a string or a char, you need to cast it to “void **” first.

If you want to define a pointer to a two-dimensional array, you need to use “int (*p)[3];” instead of “int *p[3];”. And you traverse the array by using “p[i][j]” instead of “p[i * 3 + j]”.

A pointer that points to a function is called a function pointer. For example, we can define a function pointer “int (*p)(int, int);”. And we assign a function to it by “p = add;”. Then we can use “p(1, 2);” to call the function.

Memory Allocation

In C, we have 4 functions to allocate memory:

- void *malloc(int num) - allocates an array of num bytes and returns a pointer to the first element of the array.
- void *calloc(int num, int size) - allocates an array of num elements each of which size in bytes will be size, all bytes in the array will be initialized to 0.
- void *realloc(void *ptr, int new_size) - re-allocates memory extending it upto new_size.
- void free(void *ptr) - deallocates the memory previously allocated by a call to calloc, malloc, or realloc.

For example, if you want to allocate memory for an array of 10 integers, you can use “int *p = (int *)malloc(10 * sizeof(int));”. And use “free(p);” to deallocate the memory.

In C++, we have 2 functions to allocate memory:

- new - allocates memory at runtime.
- delete - deallocates memory at runtime.

For example, if you want to allocate memory for an array of 10 integers, you can use “int *p = new int[10];”. And use “delete [] p;” to deallocate the memory.

Templates

In C++, we can use templates to create generic functions and classes. For example, we can use “`template <typename T> void swap(T &a, T &b)`”. You instantiate a template by providing a type explicitly or implicitly, like “`template void swap<int>(int &a, int &b)`”. If you want to specify the function for a specific type, you can use “`template <>`”.

Enum Type

In C++, we can use “enum” to define a new type. For example, we can use “`enum color {red, green, blue};`” to define a new type “color”. And we can use “`color c = red;`” to define a variable of type “color”. Also, “`color c = (color)1;`” and “`color c = color(1);`” are both valid.

File I/O

We use “`#include<fstream>`” to include the file I/O library.

We use “ofstream” to create a file output stream, “ifstream” to create a file input stream, and “fstream” to create a file input/output stream.

Then we associate the stream with a file by using “open()”. Especially for “fstream”, we need to specify the mode of the file, like “ios::in” for input, “ios::out” for output.

Then we check if the file is opened successfully by using “is_open()”.

After that, we can use “<<” to write to the file, and “>>” to read from the file. We can check if we have reached the end of the file by using “eof()”.

Finally, we use “close()” to close the file.