**CS 162 Worksheet 2**

1. Scope
	1. refers to the lifetime and visibility of a variable
	2. In C/C++, a variable’s scope is the {} it is declared within
	3. Main idea: A variable dies at the end of the {…} it was declared in

Ex: For the following program, answer the following questions:

1. Where does temp and num out of scope?
2. Is num still accessible in f1()? Why or why not?
3. Can you print temp outside the if? Why or why not?
4. void f1(); //function prototype
5. int main () {
6. int num;
7. cin >> num;
8. if (num > 0) {
9. int temp = 2 \* num;
10. cout << temp << endl;
11. }
12. cout << temp << endl;
13. f1();

11. return 0;

 12.}

 13.

 14.void f1 () {

 15. cout << num << endl;

 16.}

1. C++ string

In C++, the string type (i.e., class type) is provided to handle this. To use them, include the <string> library. Answer the following questions regarding C++ strings:

* 1. How to declare a string, named my\_str, and initialize it with “hello world”?
	2. How to print my\_str out?
	3. How to access each element in my\_str? For example, how to change the first element to ‘H’?
	4. How to get the length (size) of my\_str? How to access the very last element?

\*\*\*For extended learning:

In C, strings are:

* Character arrays (i.e., char mystring [64];)
* Terminated with a NULL character (i.e., ‘\0’)
* Processed using C string library (i.e., <cstring>)
1. (Pseudo) Random number in C++

In C++, (pseudo) random number generation is accomplished with the rand() function.

* rand() returns an integer between 0 and RAND\_MAX
* RAND\_MAX is an integer constant defined in <cstdlib>, the value is implementation defined

Question:

1. How could you generate a value that is either 0 or 1?
2. What else needs to be done so that rand() would generate different random numbers in each execution?
3. Functions – primary unit of code decomposition and abstraction in C/C++

A function has:

* A name
* Zero or more parameters
* 0 (void) or 1 return

Trace through the following program and answer the questions in comments.

…

int max (int, int); //function prototype

int main () {

 int x, y, mx;

 cin >> x >> y; //assume user types 5 and 7

 **//state why each of the following is a bad function call:**

 mx = int max (x, y);

 mx = max (int x, int y);

 mx = max (a, b);

 max (x, y);

 **//put the correct function call here: (mx should store the max of x and y)**

 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_;

 cout << mx << endl; **// what’s the output?**

 cout << max(0, x) << endl; **// what's the output?**

 return 0;

}

int max (int a, int b) {

 if (a > b)

return a;

 return b;

}

1. Error handling
	1. Write design and pseudocode: prompt the user for a positive integer until a valid one is provided. The user may enter anything, including strings.

Hint: stoi() from <string> is used to convert a string to an int

1. 1D static array

An array is a ***fixed size***, ***named collection*** of ***ordered*** variables of the ***same type*** that are accessed with an index and stored contiguously in memory.

– Fixed size: Cannot grow or shrink

– Named collection: One name to refer to all variables in the array

– Ordered / Accessed with an index: Individual element is accessed with its /index (using [])

– Same Type: Elements in one array must all be the same type

Ex.

1. Declare an array of 50 doubles. Follow up: What are their initial values?
2. What is the size and index range of the following array? How to access the last element?

string my\_strings[10];

1. For the array declared in b, how to print all elements?
2. How to pass an array into functions?

Coming From Other Languages?

SIMILARITIES: Like Python and Java, C/C++ arrays

* Use 0-based indexing (beginning element at index 0)
* loops can be used to iterate over all the elements of an array

DIFFERENCES: Unlike Python and Java, C/C++ arrays

* Are fixed size (size must be a constant) and then cannot grow easily after being declared
* Do not remember their size (no len() or .length) nor bounds-check an access (so

accessing array[1048726] will happily execute in C/C++ and likely cause a crash (i.e. "Segmentation Fault")

* Are NOT objects (no .append() or .length) in C/C++