Functions II

Recursion review: Place the loop after the function call if you want number to print forward (12345.

Place the loop before the function call if you want number to print backward (54321)

Variable Scope- The lifetime of the variable

Variable scope defines where a variable is visible and accessible.

The first cout requested beneath the variable top to bottom will be called first

Function Variable- one the function dies, the variables die as well.

Global function

Static variables are global variables

Dynamic memory stored in heap memory. Subject to the whim of the coder.

**Addressof** operator-memory address of variable can be output

cout << **&x** << endl;

Addresses are useful for creation of arrays with many elements.

The integer pointer points to the first element in the array.

Create a variable. Assign it a value. Create an integer pointer (**int \*x\_ptr; x\_ptr = &x;**

If a **\*** precedes a variable name, it is a pointer.

The datatype tells the pointer where to look.

**&** indicates a pointer.

Pointer address is the memory value.

Pass parameters by value you share the value.

When you pass parameters by reference, you can change the value by referencing the address.

Dynamic memory means that you can build things.

**&n** is the pass by reference syntax

endl is another stop reading command

; stop

return back to main

\n break off, start a new line

**Victor 4/4/2020**

Variable scope describes where a variable is seen and unseen.

Where a variable is declared matters.

If we define and declare a variable in a main, it is only seen in the main.

If we define and declare a variable in a function, it is only seen in the function.