Jan 23rd

* Declaring pointer
  + Int \*p;
    - Int a = 2;
    - P=&a; (P points to a)
    - Cout << \*p; (dereferences the pointer printing out the content of the pointer)
  + Double \*ptr;
    - Double l;
    - Ptr = &l;
* Array (which is a pointer)
  + Double list[10];
  + Stores 10 consecutive doubles in memory
  + Then there is one more spot reserved for a pointer with the name of the array that to the beginning of the array.
  + For(int I = 0; I < 10 ; I++)
    - List [I] = I+1; (sets each spot equal to its position)
  + Double \*p;
    - P=list;
  + For (int I = 0; I < 10; I++)
    - \*p = I+1
    - P++; (points to the next memory location in the array)
* The name of the arrary is a pointer to the beginning of the array but it will not change. It is a constant.
  + P = list;
  + \*p = 1;
  + For (p = list + 1; p < list+10; p++)
    - \*p = \*(p-1) +1; (each time through the loop, p sets the next location to plus 1 of the previous content.
* C Char strings (different from c++ strings)
  + Are null terminated
  + Array of chars.
    - Char str[10] = "test"
    - In memory you get 10 bytes for chars. Memory location 0 – 3 is t-e-s-t. Then location 4 is '\0' (null character).
  + Function to find length of the array
    - Int strlen(char \*s)
      * Int I ;
      * For (I =0; s[I] != '\0'; I++);
      * Return I;
  + Everything is true except for 0.
    - S[i]!='\0' is equal to just s[I]
  + Changing the function to pointer
    - For (I = 0; \*s; I++, s++);
  + Making it even shorter
    - For (I = 0; \*s++; I++); ++ is a post fix operator. So itll return \*s then increment it
  + For example
    - Int I =1, j = 2;
    - I = j++; so I would become 2 then j would increment to 3
    - But if I = ++j; would increment j to 3 then assign to I;
* String copy
  + Void strcpy(char \*t, char \*s) (s is source and t is target. Right to left)
    - For (; \*t = \*s; t++, s++); (No need to initialize because the pointers start at the beginning of the arrays. Sets t's content to s's content then increments their locations).
    - Or
      * For (; \*t++=\*s++;); (simpler way to write it) (probably harder to read though lol)
    - Main()
      * Char str1[10] = " test, str2[10] = "abcde";
      * Strcpy(str2, str1);
      * Cout << str2;
* String compare.
  + Abd < abcd = true
  + Abd > abcd = true because d is > then c in position 3.
  + Abc == abc = true
  + AB < abc = true because capitals are fist on the ascii table
  + Int strcmp(char \*t, char \*s)
    - For (; \*t++ == \*s++&& \*t;);
    - Return \*t-\*s;
* Ch 7, C++ strings
  + Defined in STL. (standard template library)
  + #include <string>
  + String s = "abcd", t="ef";
  + Cout << s.length();
  + S+=t; concatenates. Attaches t to the end of s.
  + (s.operator+=(t);) is what happens when we call that += operator.
  + S=t; overwrites the value of s with the value of t;
  + If (s>t)
  + String u (s); Makes a new string u that copies s. //Copy constructor.
  + Which is the same as String u =s;
  + u.insert(2, "lmn"); Inserts lmn at position to. Abc -> ablmnc.