Notes about C

* %d, %f, %s are format specifiers for decimal, float, string respectively (used when printing)

int age = 18;

char name []= “Mary”;

//could also be char\*name = “Mary”; - this is called pointer notation

//if you want to specify the length of a string you can but don’t have to 🡪 if you do, must add an extra value (terminator value, is a zero but doesn’t print)

printf (“%s is %d years old.”, name, age);

* /n means add a new line

printf(“%d\n”, age);

* strncmp is a function that compares strings – takes two strings and a maximum comparison length as arguments, will return zero if they are equal, otherwise will return a different number

char\*newName = “Adam”;

if (strncmp(newName, “Adam”, 4) == 0) {

printf(“Hello, Adam”);

}

else {

printf(“You are not Adam.”);

}

* For loops
  + \*must initialize “i” outside of the for loop

int i;

for (i = 0; i < 10; i++) {

printf(“%d\n”, i);

}

* While loops
  + Break: halts a while loop
  + Continue: allows some part of code to be skipped if a certain condition is true

while (n<10) {

n++ //1 added each time no matter what

if (n%2 != 0) {

continue; //skips back to top of loop

}

printf(“%d\n”,n);

}

* Methods in C
  + Method headers are simpler:

int multiply (int x) {

//could also be void instead of having a return type

return x\*2;

}

* + Example:

int printBig(int x) {

if (x > 10) {

printf(“%d is big\n”, x);

return 0;

}

else return 0;

}

int main () {

//main method

int numbers [] = {1,2,3,4,5,6,7,8,9,10};

int i;

for (i = 0;i<10,i++) {

printBig(numbers[i]);

//call the method

}

}

* Pointers: simple integer value that holds a memory address that points to a value INSTEAD OF holding the value itself (like a class instead of a primitive type)
  + Dereferencing: refers to where a pointer points at instead of the actual address that holds something 🡪 \* is the dereferencing operator, must be used when creating and referencing a pointer

int a = 1;

int\*pointerToA = &a;

//use asterisk to define a pointer variable, as well as the & sign

//now this variable points to a

* Structures: large variables that contain several variables inside, including pointers

typedef struct {

int year;

char\*model;

//can contain a string

} *name;*

*name* x;

x.year = 1988;

x.model = “Toyota”;