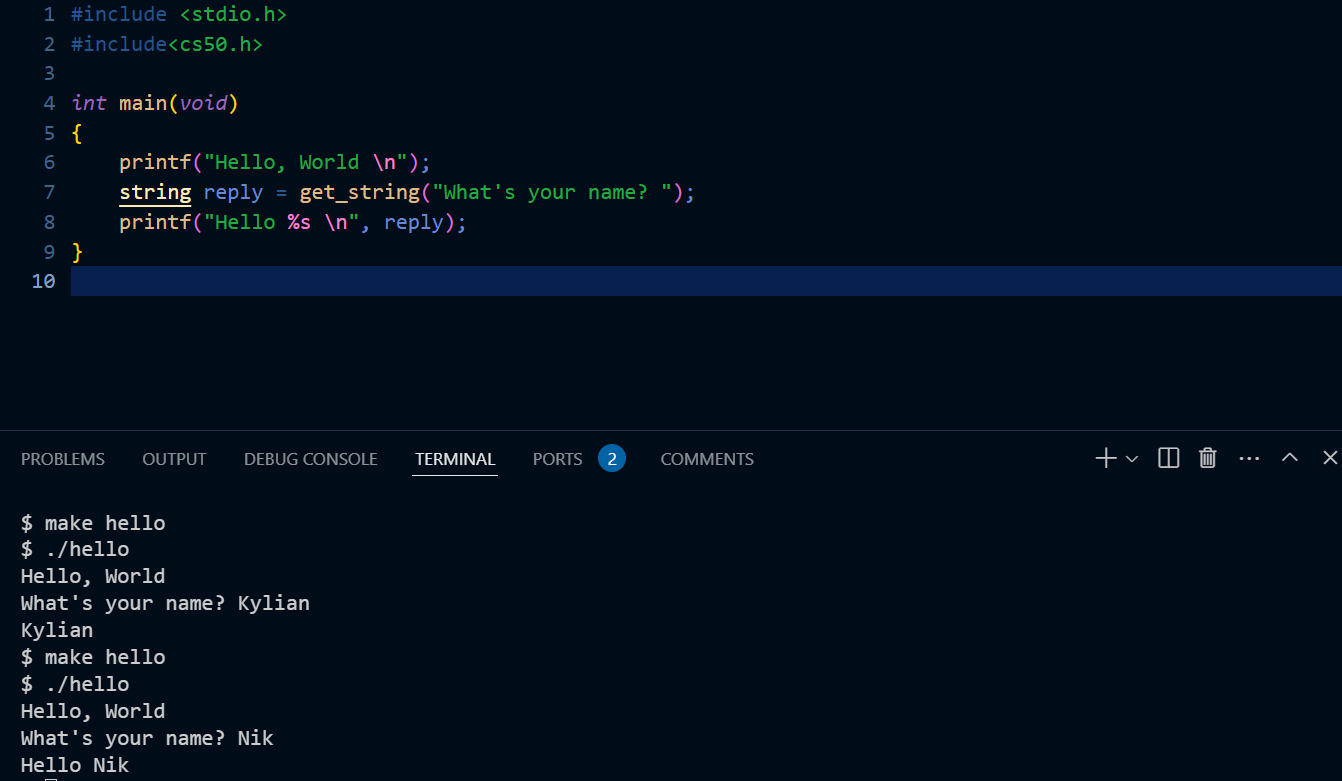
Machine Language:

1. Humans write “source code”
2. Computers only read binary, aka “machine code”
3. In the past, we used to use punch cards with holes inserted at different points to feed data to a computer.
4. If our input is the source code, and the output is machine code, then the algorithm which performs the conversion is the compiler

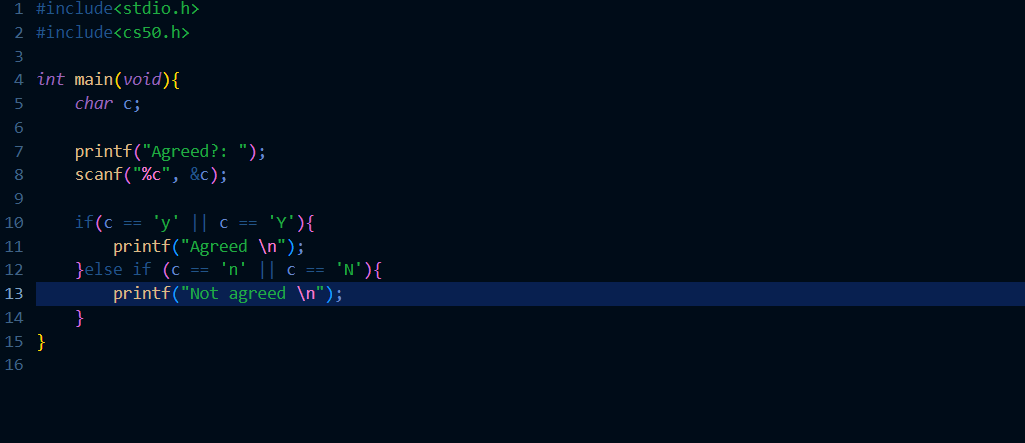
Hello World:

1. code filename.c – Creates a file with the name filename
2. make filename – This command will make the C source code into machine code
3. ./filename – This command lets you run your code
4. So the steps would be to create the file, then perform the conversion using the make command, and finally running the code itself.
5. By convention, always name your files in lowercase. Never use spaces while naming files.
6. The f in printf stands for formatted.
7. \n is the new line character. It is known as an escape sequence

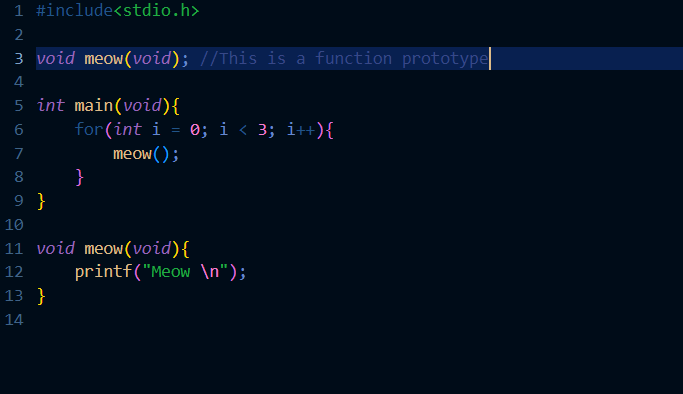
CS50 Library:

1. When we say #include, essentially we are telling the compiler “Hey, there’s a file on the local hard drive that you need to go find, once you do, paste it here”.
2. Stdio.h is called a header file. It is a library. A library is code someone else wrote, that you can use.
3. manual.cs50.io – this website gives documentation of C.
4. “%s” is to output strings
5. Parameter = the placeholder for the function, to let it know we’re gonna pass something to it
6. Argument = The actual value that we pass
7. 
8. Note: get\_string is part of the cs50 header file, we can’t use it in the piscine
9. Note: scanf can only be used for receiving input. If we wanted to write “What’s your name?”, then we need to use a printf statement before it.
10. Note: the string data type is also part of the cs50 header file. We’ll need to use character arrays.
11. For formatting reference : “%s” for strings, “%i” for integers and “%c” for characters

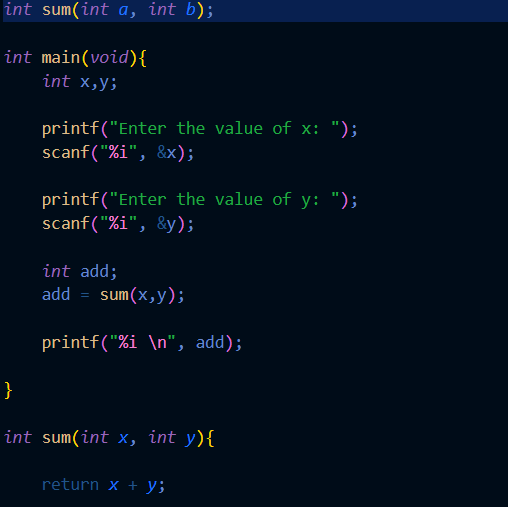
Conditionals and variables:

1. Note: counter = counter + 1 is the same as counter += 1 and counter++
2. When we type ./filename to run the program, the ./ means that the file is in the current folder
3. scanf requires the format and also a memory address of the variable where it should put the value that was read. So we need %i and also the memory address symbol &
4. Note that all C strings end in \0. This in turn can cause a lot of buffer overflow issues, which in turn can cause hackers to exploit poorly written code.
5. When entering only a single character, use single quotes. When entering a string, use double quotes.
6. 
7. The above is the use of the OR operator in C
8. Code can be evaluated upon three axes. First, *correctness* refers to “does the code run as intended?” Second, *design* refers to “how well is the code designed?” Finally, *style* refers to “how aesthetically pleasing and consistent is the code?”
9. A compiler will convert the source code to machine code. A transpiler will convert one language to another language, like Babel changes JSX to JS

Functions:

1. The syntax is:
   1. void functionname(void)
   2. { printf(“Hello”); }
2. The first void shows that we aren’t returning anything, the second void shows that we aren’t passing any arguments to the function
3. 
4. Use function prototyping in general. This is because of 2 reasons:
   1. If we have a lot of functions, then main() will keep on moving downwards. Also, function A might need to be above function B, and function B needs to be above function C, **but function C may need to be above function A**.
   2. Also, it’s best practice to have main() at the top, so people can easily find it.

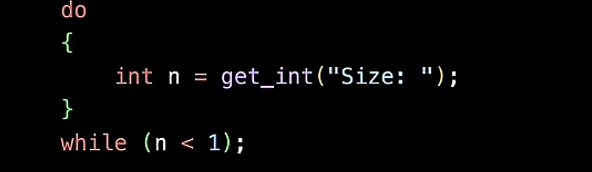
Scope:

1. 
2. The name of the variables in the parameter position of the function declaration does not have to be the same as the ones in the actual function definition. (See function sum)
3. The main function returns the number 0 if everything ran correctly, and it returns any other number if it fails.

Linux commands:

1. Popular commands include cd, cp, ls, mkdir, mv, rm, mkdir
2. The mv command allows you to rename a file.
   1. Example: mv originalFilename newFilename

Constants:

1. We can make a variable a constant, i.e, it cannot be changed in any other part of the program by using the keyword const.
   1. Example: const int x = 5;
2. 
3. The above loop asks the user to enter an integer input above 1. If the input isn’t int, then the get\_int function would ask for a new input. And if the input is below 1, then the do while loop keeps running. How would we do this without the cs50 function?

Memory and integer overflow:

1. Imagine we have only three bits, and we want to represent 7.
2. We would do so as 111.
3. Now imagine we want to add 1, and make it 8.
4. When we add 1 to 111, we should get 1000. However, since we only have 3 bits, we will get (1)000, where the 1 is discarded. So ultimately, 8 is represented as a 0. This is integer overflow.
5. An int type is of 4 bytes = 32 bits. So we can store up to 2^32 which is approx. 4 billion 300 million something.
6. If we include negative numbers, then the above number is halved.
7. The long data type is 8 bytes = 64 bits.
8. Truncation – when you divide an integer by an integer, then decimal part just gets thrown away, even if you store it in a float. This is because you are only doing integer based math.
9. We need to use type casting in order to get the appropriate float answer.
10. “%.5f” tells printf to show us up to the 5th decimal place.
11. Remember that due to the rounding errors, maths using Java or C is imperfect.