***Lecture 0***

**Computer Science and Problem Solving**

* Input -> [] -> Output
* Computers count using the binary system. The binary system uses 2 digits: 0 and 1. A bit is named from the term binary digit. A bit is 0 (an off) or 1 (an on).
* Using binary code and three lightbulbs:
  + 4 2 1

0 0 0 = 0

* + 4 2 1

0 0 1 = 1

* + 4 2 1

0 1 0 = 2

* + 4 2 1

1 0 0 = 4

* + 4 2 1

1 0 1 = 5

* + 4 2 1

1 1 1 = 7

* Adding another bit would let the binary code go up to 8
  + Like this: 8 4 2 1

1 0 0 0 = 8

* There’s 8 bit’s in 1 byte. Computer’s use eight bites to represent a number:

128 64 32 16 8 4 2 1

0 0 0 0 0 0 0 0 = 0

**ASCII**

* **For letter A the binary code is 65:** 128 64 32 16 8 4 2 1

0 1 0 0 0 0 0 1 = A

* **The chart for the English language:**

| **0** | **NUL** | **16** | **DLE** | **32** | **SP** | **48** | **0** | **64** | **@** | **80** | **P** | **96** | **`** | **112** | **p** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | SOH | 17 | DC1 | 33 | ! | 49 | 1 | 65 | A | 81 | Q | 97 | a | 113 | q |  |
| 2 | STX | 18 | DC2 | 34 | ” | 50 | 2 | 66 | B | 82 | R | 98 | b | 114 | r |  |
| 3 | ETX | 19 | DC3 | 35 | # | 51 | 3 | 67 | C | 83 | S | 99 | c | 115 | s |  |
| 4 | EOT | 20 | DC4 | 36 | $ | 52 | 4 | 68 | D | 84 | T | 100 | d | 116 | t |  |
| 5 | ENQ | 21 | NAK | 37 | % | 53 | 5 | 69 | E | 85 | U | 101 | e | 117 | u |  |
| 6 | ACK | 22 | SYN | 38 | & | 54 | 6 | 70 | F | 86 | V | 102 | f | 118 | v |  |
| 7 | BEL | 23 | ETB | 39 | ’ | 55 | 7 | 71 | G | 87 | W | 103 | g | 119 | w |  |
| 8 | BS | 24 | CAN | 40 | ( | 56 | 8 | 72 | H | 88 | X | 104 | h | 120 | x |  |
| 9 | HT | 25 | EM | 41 | ) | 57 | 9 | 73 | I | 89 | Y | 105 | i | 121 | y |  |
| 10 | LF | 26 | SUB | 42 | \* | 58 | : | 74 | J | 90 | Z | 106 | j | 122 | z |  |
| 11 | VT | 27 | ESC | 43 | + | 59 | ; | 75 | K | 91 | [ | 107 | k | 123 | { |  |
| 12 | FF | 28 | FS | 44 | , | 60 | < | 76 | L | 92 | \ | 108 | l | 124 |  |  |
| 13 | CR | 29 | GS | 45 | - | 61 | = | 77 | M | 93 | ] | 109 | m | 125 | } |  |
| 14 | SO | 30 | RS | 46 | . | 62 | > | 78 | N | 94 | ^ | 110 | n | 126 | ~ |  |
| 15 | SI | 31 | US | 47 | / | 63 | ? | 79 | O | 95 | \_ | 111 | o | 127 | DEL |  |

* Because binary code can only count up to 255, there is a limit to how many letters that there are in ASCII

**Unicode**

* As time continues, there are more ways to communicate using text.
* Since there wasn’t enough digits in binary code to express all the different characters, the Unicode language had to expand the number of bits that are being transmitted through computers.
* Emojis are being made using that binary code, using bits. Different patterns make different emojis.

**RGB**

* Zeros and ones are used to make color too.
* RGB (Red, Green, and Blue) require a combination of three numbers to make one color.
* The binary code can be used to make videos, audio, and pictures as well.

**Algorithms**

* Problem solving is a central to computer science and computer programming. An algorithm is a step-by-step set of instructions to solve a problem.
* Finding a name in a 1000 page phone book can be easy if you divide the number of pages by 2 each time, but the problem is you might go past the name in the phone book if you go through the book too fast.

**Pseudocode**

* Using functions (an action or verb) you can make a set of code to work on something
* Else, Else If, and If are called conditions.
* The questions that you ask with the conditions are called “*boolean*” expressions. A Boolean expression is a question with a yes or no answer.
* Large language networks are implemented using something called neural net works.
* Having a rubber duck on your desk or wherever your working is a good way programmers figure out their problem. They tell the duck about their problems which brings a solution to their problem. That is called rubber ducking.

**Coding**

* In coding using the letter “n” is for describing the number of times that the function will happen.
* Using the codes you can program the “sprite” on the right to do actions, like talking or moving.

***Lecture 1***

* There are three parts to a compiler: the area where the explorer is, the right top where the commands are, and the bottom right where the terminal is where the commands are executed.
* Common command lines are:
  + cd – which is used for changing the current folder
  + cp – used for copying files in a directory
  + ls – used for listing files in a directory
  + mkdir – used for making a directory
  + mv – used for moving or renaming files and directories
  + rm – used for removing/deleting files
  + rmdir – also used for removing/deleting files

**Hello World**

* There are three commands to write, compile, and run a program:
  + Code hello.c
    - Creates a file and allows to type instructions for the program
  + make hello
    - Compiles the file from the instructions in C and makes an executable file called “hello”
  + ./hello
    - This command runs the program called “hello”

**From Scratch to C**

* The “say” block in scratch is used to display any text on screen, but for C, the code “printf” is used for the exact same reason.
* Any statement of code should be finished with a semicolon.
* There are 6 different escape codes you can use:
  + \n – used to create a new line
  + \r – used to return to the start of a line
  + \” – used to print a double quote
  + \’ – used to print a single quote
  + \\ - used to print a backslash

**Types**

* There are 7 different data types:
  + get\_string – used to create a string of text
  + get\_bool – used to determine true or false
  + get\_char – single characters instead of full phrases
  + get\_double / floats – real numbers something with a decimal point also the equivalent to fractions
  + get\_int – variables that are designed to hold integer values
  + get\_long – basically integers but longer

**Conditionals**

* The difference between the Scratch and C code is that whenever trying to represent the hugging shape in scratch it’ll look like this: A screenshot of a computer

  AI-generated content may be incorrect. but in C it’ll look like this: A computer screen with white text

  AI-generated content may be incorrect.. The difference is in the brackets.

**agree.c**

* When using strings you use double quotes but when using single chars you use single quotes
* When getting the return value, and want to check what the one char is, you need to put the char that you are comparing against in single quotes or apostrophes both for the y and n.
  + A computer code on a black background

    AI-generated content may be incorrect.

**Loops**

* For loops, sketch uses the repeat command, but for C, there is a different command used to loop what you want to loop. A screenshot of a phone

  AI-generated content may be incorrect.A black background with white text

  AI-generated content may be incorrect.

**Correctness, Design, Style**

* Three different codes help with fixing your code:
  + check50: it will check the correctness of your code
  + design50: there is a place to use this and that place will give you advice on how you can make your code look better
  + style50: it also helps with making your code look better

***Lecture 2***

**Compiling**

* Encryption: the act of hiding plain text from the human eye
* Decryption: the act of taking a piece of encrypted text and making it readable

**Debugging**

* One debugging technique is called “rubber duck debugging”, where you can talk to an inanimate object or yourself to help you think through the problems or difficulties in your code.
* Each data type requires certain amount of system resources:
  + bool – 1 byte
  + int - 4 byte
  + long - 8 byte
  + float - 4 byte
  + double - 8 byte
  + char - 1 byte
  + string - ? byte
* Arrays are sequences of values that are stored back-to-back in memory
* “int scores [3]” is a way of telling the compiler to provide three back-to-back places in memory of size “int” to store three “scores”.
* A string Is an array of variables of type ”char”: an array of characters?

**Command-Line Arguments**

* Command-Line Arguments are arguments that are passed to your program at the command line.

The rest of this category is just showing what to put into the command prompt

**Exit Status**

* When a program ends, a special exit code is provided to the computer.
* When the program exits without an error, a status code of “0” is provided to the computer. But when there is an error while exiting the program, the status code of “1” is provided to the computer.
  + Typing “echo $?” in the terminal will let you see the exit status of the last run command.

**Cryptography**

* Cryptography is the art of deciphering and ciphering a message
  + Now that you have the building blocks: arrays, chars, and strings, now you can cipher and decipher a message.
  + “plaintext”, and a “key” are used in a cipher, which then result into a deciphered message.

A black and white square with white text

AI-generated content may be incorrect.

(I used this from the notes slide because I couldn’t get a good shot from on the video)

If you input the right letters and number it would look like this:

A black square with white lines

AI-generated content may be incorrect.

***Lecture 3***

**Linear Search**

* An array is blocks of memory that are consecutive: side by side with one another.
  + Like: A red rectangular object with black dots

    AI-generated content may be incorrect.
  + The far left is called “location 0” and the far right is called “location 7” the end of the array.

**Binary Search**

* Binary search is another search algorithm.
  + There wasn’t anything else talked about in this sections of the video.

**Running Time**

* Running Time is using an analysis using big “O” notation. Like in the graph below.
  + A graph with a green line

    AI-generated content may be incorrect.
  + A screenshot of a computer

    AI-generated content may be incorrect. <- these are some common running times (the way an algorithm’s execution time grows as the input increases) in the algorithm.
  + The Ω symbol is used to represent lower bound while the O symbol is used to represent the upper bound. But if the upper bound and the lower bound is the same then you use the Θ symbol. Generally the Θ is not used but sometimes the algorithm performs at the upper and lower rate at the same time which makes it use the Θ symbol.
  + A screenshot of a computer

    AI-generated content may be incorrect.The place where the O symbol is can be exchanged with the Ω and Θ symbol.
  + *Asymptotic Notation* is a notation that you use when talking about the running time of algorithms over time. Also, how you measure its efficiency.

**Sorting**

* Sorting is when you take an unsorted list of values and transform it into a sorted list.
  + When a list is sorted, searching the list will be less hard to find something from on the computer.
    - You can use binary search to search on a sorted list but you can’t use it on a unsorted list.
    - Another sorting algorithm is called *Selection Sorting.* There is an algorithm for it. A black background with white text

      AI-generated content may be incorrect.
* There is another sorting algorithm called *Bubble Sorting*.
  + The code for this algorithm is: A black background with white text

    AI-generated content may be incorrect.

**Recursion**

* Recursion is a concept in programming where a function calls itself.

***Lecture 4***

**Pixel Art**

* Pixels are dots that use color and are arranged in a way to make some piece of art.

**Hexadecimal**

* Hexadecimal is a system of counting that has 16 counting values.
  + The values are: 

**Pointer Arithmetic**

* Pointer Arithmetic is the ability to do math on locations of memory

***Lecture 6***

* Filter
  + Typing “code blur.py” imports two modules, “Image” and “ImageFilter” from the library called “PIL”. This takes an input file an creates an output file.
* Functions
  + The function “printf(“hello, world\n”);” is how it looks in C, but in Python, the same function looks different, like “print(“hello, world”)”. The difference is in C there is an “f” after print and there is “\n” after “world”, but in python those aren’t used in the code.
* Libraries, Modules, Packages
  + In C, you can import the CS50 library, you can also import the library on python as well.
    - The code to import the library is: “import cs50”
    - To import specific functions in the cs50 library use code: from cs50 import get\_float, get\_int, get\_string”
* Data types
  + In python, the amount of data types get cut in half. In C there are 7 different data types: bool, char, double, float, int, long, and string. Meanwhile, in python the data types are: bool (used for capital true, capital false values), float (used to represent real numbers using a decimal point), int (used to represent integers, and str (short version of string)
* Calculator
  + Creating a simple calculator on python is easier than C because on C you must include the function “f(“%\n” between the code of “printf(“%i\n”, x + y);”
* Conditionals

A screenshot of a computer screen

AI-generated content may be incorrect.A computer screen with white text

AI-generated content may be incorrect.A black background with white text

AI-generated content may be incorrect.

* + When writing the code: in C, it is the exact same but there are brackets to replace the distance between each print function. There also is an ”else” function for the second line of ”print”. But in python, the brackets and the semi-colon are gone.
* Loops
  + Using the “range” function is the same as using the repeat block in scratch. You put the range function and in parenthesis you put the number that you want the range to go up to. For example, you put “for i in range(5)” in the command prompt, that makes the counting start from zero and count to 5.
  + Sometimes you can put an “\_” in place for the letter “i" in the function “put i in range(#)”,
* Lists
  + Using the command “len” lets you see the length of a list. When in C you have to keep track of the length of the list but in python or other programming languages, you can just type out the len command and it will tell you the length.