**Intro to Comp Science Notebook**

**Unit 1: Computing Systems**

Chapter 1: What is computer science?

* 5 concepts of computer science
  + Computing systems (hardware)
  + Algorithms and computing
    - Program – a set of commands and instructions translated into code for a computer to understand
    - Algorithm – list of steps or instructions to complete a task
  + Data and Analysis
    - Analysis – organizing, describing, and understanding
  + Networks and Internet
  + Impacts of computing

Chapter 2: What is a computer?

* Software – a set of programs (instructions) that tell a computer what to do like phone apps or games
* Input – parts that send info to a computer such as keyboards and webcam
* RAM – random access memory
* CPU central processing unit– brain of the computer that receives input info and executes commands from hardware and software
* GPU graphics processing unit – delivers images, animations and videos viewed on computer screen
  + CPU is for general info processing and GPU is specified for graphics and images
* Operating Systems – programs that communicate with hardware allowing other programs to run in the background
  + Makes it possible to install games, social media and other apps
  + Ensures that individual pieces of hardware run with all different programs – integrates apps and hardware and software

Chapter 3: Interacting with Computers

* GUI – graphical user interface – interface that uses icons and symbols on a screen rather than plain text to interact with a computer or programs
* Command Line Interface – uses text only to operate a computer – not user friendly but the operator can be more specific and precise with its commands to the computer
* Troubleshooting – systematic approach to solving errors
* Systemic approach – following a fixed or step by step process to solving a problem
* Debugging – finding and correcting errors within a program

**Unit 2: Data and Analysis**

Chapter 4: Data

* Binary – 1s and 0s that computers understand, it can be a grid to create an image, colors, numbers etc. and depending on the type of file, the computer will know what to translate it into
* RGB – color notation that uses three numbers separated by commas, where each number is between 0 and 255, each of the three numbers represents a different shade of red, green, blue
* Hexadecimal uses 6 characters to represent color such as FF00B4

Chapter 5: Collecting and Using Info

* NA

**Unit 3: Software Engineering**

Chapter 6: Designing Computer Systems

* Engineering – a branch of science that studies the design, building, and use of machines and structures to solve a problem
* Engineering design process
  + Identify the problem
  + Plan a solution
  + Build/make a plan
  + Test
  + Improve
* Chapter 7: Testing
  + NA
* Chapter 8: Documenting
  + NA
* Chapter 9: Feedback
  + NA
* Chapter 10: collaborating
  + Libraries give access to helpful premade functions in a programing language such as Pandas for working with data in Python

**Unit 4: Algorithms and Programing**

Chapter 11: Using Algorithms

* NA

Chapter 12: Programming Languages

* Each programming languages has their own strengths and weaknesses
  + Javascript, HTML, CSS - primarily front end language
  + Python, C++ - primarily back end languages that connect and organize front end info
* C++ is faster at running programs but Python is easier to write. So the initial code might be written in Python but then later replaced with blocks of C++ for performance
* Binary - computers process info using electrical circuits that can turn on at off or 1 and 0 - this language is called binary or machine code
* the CPU of a computer can only read instructions written in binary so after a human writes the code (Python, Java, C++) then that program changes it into binary - this is called COMPILING
* Bit = 1 binary digit, 8 bits = 1 byte
  + the terms like giga and mega bytes refers to quantity of binary digits can be stored
* Kilobyte - 1,000 bytes ex a 5 page Word doc is about 100kb
* Megabyte - 1,000,000 bytes ex an audio file is several MBs
* Gigabyte - 1,000,000,000 (billion bytes) ex an HD movie is a few GBs
* Teratype - 1,000,000,000,000 (trillion bytes) Hubble space telescope sends about 10TB of data every year

Chapter 13: Computational Thinking

* 4 Major Areas
  + Decomposition - breaking the problem down into smaller parts
  + Pattern recognition - ID what different problems have in common
  + Abstraction - separating details that matter from those that don’t
  + Algorithm Design - creating a solution with simple steps that can be used widely

**Unit 5: Universal Programming Principles**

Chapter 14: Variables

* NA

Chapter 15: Conditional Statements

* NA

Chapter 16: Loops

* Loops - allow programmers to repeat a chunk of code many times such as a ball bouncing up and down in a game
* A loop makes code blocks shorter, sames programmers time, and is more efficient for the computer to run
* For Loop - repeats itself a set number of times and used when you know the number of times you want to repeat something
  + EX Top10 = [player1, player2, player3, player4,...]
  + For item in top10  
     print(item)
  + This will print all the names in the top 10 players list above without having to write print(player1), print(player2) etc.
* While Loop - will run a chunk of code until a condition is met and used when you don’t know the exact number of times you want the loop to repeat
* EX to keep asking a question until the player gets a question right, it could be on the first attempt or it could be on the 100th, the code of asking a question will keep running until one is answered correctly, while a question is answered wrong, keep asking a question

Chapter 17: Events

* Events are actions that cause something to happen within a program
  + Can be external (pressing the spacebar
  + Can be internal (like a web browser loading a page)
* An event handler is the code that is run when an event happens
  + ex on press\_spacebar  
     swing chef’s knife

Chapter 18: Procedures

* Procedure (similar to a function) is a piece of code that has a name and completes a specific task
  + EX instead of programming each of the 4 players to jump when the jump button is pressed, create the one jump procedure and apply it to each of the 4 players
* Parameters - variables passed into a procedure and can only be used within that procedure

**Unit 6: Programming in Scratch**

Chapter 19 - 23: High level overview

* Scratch - free graphical programming language that uses drag and drop blocks with precoded commands that “snap” together to imitate a real programming language

**Unit 7: Programming in Python**

Chapter 28: Python Lists and Boolean

* List – variable that stores multiple values of all different data types
  + Fruits = [‘apples’, ‘bananas’, ‘oranges’, ‘pears’]
* Indexing – used to find or change certain values in a list
  + Indexing always starts with 0
  + Indexing is also exclusive of the last index number in a range
  + Ex print(fruits[1:3]) will only print bananas and oranges and NOT pears which is item 3
* If there is a list within a list, then that inner list is treated as a single item within the full list
* = is used to assign a variable, == means ‘is equivalent to’

Chapter 29: For Loops

* For Loops repeat a sequence of commands and code, a specified number of times
  + Such as counting down from 10, or repeating a specific phrase a set number of times
* Steps for creating a For Loop
  + 1. Enter the key word “for” to indicate the start of the loop
  + 2. Name the counter variable whose value increases each time the loop repeats – such as if you are counting from 0-10 by 1 each time the loop starts over, the counter variable will represent 0 in the first run, then 1, then 2 etc. Many programmers tend to use ‘i’ as this variable
  + 3. Add the key work ‘in’ to show that you are about to specify how many times the loop should repeat
  + 4. Set the number of times the loop should repeat usually by using the range() function. This function is also non-inclusive of the last number given as the parameter
  + 5. End the loop with a :
* Ex:
  + For i in range(3):

Print(“hip hip hooray”)

* This means print hip hip hooray 3 times, the i will increase with each loop. First time will be hip hip hooray 0, then hip hip hooray 1, then hip hip hooray 2 and stop
* Ex 2: using for loop with a list
  + For i in range(6):

Print(fruits[i]

* + This will print all 6 items in the fruits list
  + Another way to just print everything is
    - For i in fruits:  
       print(i)

Chapter 30: While Loops

* While loops repeat as long as the conditional statement within them are True
  + Looks similar to a For loop but replaces the counting portion with a conditional statement – continue running this code while this condition remains true
  + General format is starting with while > Boolean expression > end with : > intent the code to be repeated
* EX. Create a password checker that will continue to loop and ask to input the correct password.

Password = none (creating the variable before the loop for the loop to use and store)  
while password != ‘myPassword123’:  
 password = input(‘enter the password: ‘)  
 if password != ‘myPassword123’:  
 print(‘your password is incorrect’)  
 print(‘correct Password’)

* This look will continue to ask the user to input their password and as long as it does not equal ‘myPassword123’ the loop will run again and ask to input the password
* Once the input does equal ‘myPassword123’ then the second print will run ‘correct’ and the loop will cease
* To exist an infinite loop > Ctrl+C

Chapter 32: Functions

* You can import premade functions by importing libraries using the import key word
* Defining your own function
  + def hello():  
     print(“hello world”)
  + call the function with the name and ()  
    hello()
  + this will print “hello world”
* Defining functions with parameters and return values
  + Def convert(meters):  
     feet = meters \* 3.281  
     return feet
  + Convert(1) will return 3.281 meaning 1 foot is 3.281 meters
* Note that you can only use the return output outside of the function
* So you can do print(feet) but not print(meters)