Computer Science 1026:

Lecture #1:

- Definitions/Code:
  - IDE = Integrated Development Environment. (Ex – pyCharm)
    - Helps the interpreter by listing line numbers of code, coloring and debugging.
  - Interpreter = (Ex – Python).
- Concepts:
  - CPU – has a control unit (coordinates input/output devices) and an arithmetic logic unit (does calculations).
  - Storage:
    - Primary Storage – stores data as long as it has electricity. (Fast/Expensive)
    - Secondary storage – stores data even without electricity. (Slow/cheaper)
  - Python was designed to have simpler and cleaner syntax, and run programs quickly, while easier to modify.

Lecture #2:

- Definitions/Code:
  - int = positive or negative whole numbers.
  - float = decimal values.
    - (Ex = 0.5, 1E6, 2.5E-2).
  - ** = used to calculate an exponent.
  - / = division that gives a float remainder.
  - // = division that gives an integer, discarding any extra float value.
    - (Ex – 1.75 = 1).
  - % = gives the remainder that doesn't go into the divisor.

        - (Ex – 172 % 10 = 2).

    - abs = returns an absolute value.

    - Sqrt = must be imported (from math import sqrt)

    - \n = goes to the next line.

    - String modifiers = % (before other modifiers), then a number for places (negative starts it at the start) the number shows the amount of places the variable or word will occupy.) and then .a number for decimals, and lastly a letter (f for float, s for string).
        - (Ex - if price = 17.29463 print("%10.2f %(price)) gives _|_|_|_|_17.29).

- Concepts:
    - Constant – are typically named with capitals.
        - (Ex = BOTTLE_VOLUME).
        - Python lets use change constants, you just shouldn't.

    - Converting a number from a float to an integer rounds down.

    - Objects – are software entities that represent values with certain behaviors (Ex – a string or an int or as complex as a data file.

    - Methods – a collection of instructions to carry out a specific task. Control the behavior of objects.

    - Functions – are standalone operation's and generalizable.

Lecture #3:

- Definitions/Code:
    - if statements = always end with the equals sign if they have one
        - (Ex - >=).

    - Lexicographical Order = Space > Numbers > Uppercase > Lowercase Letters.

    - .endswith() = returns true if a string ends with the given parameters.
        - (Ex - .txt).

- .startswith() = returns true if a string starts with the given parameters.
  - (Ex – cs1026).

- .count() = returns the number of occurrences with the given parameters.

- .find() = returns the first index where the string has the given parameters, or -1 if not found.

- .isalnum() = returns True if the string has only letters/digits.

- .isalpha() = returns True if a string contains only letters.

- .isdigit() = returns True if a string only contains digits. (A negative sign doesn't count as a digit).

- Concepts:
  - Compound statements have a colon at the end.
    - (Ex – "if" statements have ":").

  - Note that float values are automatically rounded at a certain level of precision so this needs to be accounted for
    - (Ex - in python 2 == sqrt(2) * sqrt(2) is false, as it results in 2=2.000000000000004).

  - Epsilon used to compare if a difference is close enough to be negligible.
    - (Ex – if abs([sqrt(2) * sqrt(2)] – 2) < EPSILON:).

Lecture #4:

- Definitions/Code:
  - Sentinel/Flag Value = a special value to signal the last item in a list when you don't know how many items it has.

  - \n = in a print statement goes to the next line.

  - , end=" " = attaches the next print statement to the previous one.

- Concepts:

- For loops – can have up to 3 arguments. If only 1 argument it goes from 0 till that argument-1. If it has 2 arguments it goes from the first argument till the second argument-1. If it has 3 arguments it goes from the first argument till the second argument-1, by the 3d argument as an increment.

Lecture #5:

- Definitions/Code:
  - Global Scope = variables that aren't local to a function but can be used throughout the code.

Lecture #6:

- Definitions/Code:
  - Immutable = cannot change the characters in the sequence.
  - [] = a list. Is mutable. A list can also be used as a 2D table.
    - (Ex – listA = [ ).
      [0, 1, 0],
      [2, 0, 3],
      [0, 4, 0],
      ]
    - To access this list, you can use (Ex – listB = listA[2][1] to give 4). The first parameter goes down as the primary, and then the second parameter looks at the index in the first parameter.
  - .index() = yields the index of the first match of the given parameters.
  - .pop() = removes the element at the given parameter and moves up the elements lower in the list.
  - .sort() = sorts a list from smallest to biggest.
  - .list() = function that can make a copy list of another list.
    - (Ex – listA_Copy = list(listA)).
  - () = a tuple. An immutable version of a list. The most common use of tuples is to return multiple values as a single set.

- Concepts:
    - Python lists are unique in that they can use negative subscripts to access the list in reverse order.
        - (Ex – [-1] accesses the last element in a list. If the list was out of 10 for example, then [-10] would access the first value.)

    - Concatenation of 2 lists combines to lists (but doesn't pair them).
        - (Ex – list1 = [a, b] and list2 = [1, 2, 3]. Therefore list3 = list1 + list2 gives [a, b, 1, 2, 3]).

    - Replication of lists can be done by multiplying a list by a value.
        - (Ex – listX = [A, B] * 3 would give [A, B, A, B, A, B].

    - Slicing a list – is the process of obtaining only a section of the code form a list.
        - (Ex – listA_segment = listA[5 : 10] which obtains index values from 5-10). You can also slice from a certain value to the start or beginning of the list. (Ex – listA_segment = listA[5 :] or listA_segment = listA[: 7]).

Lecture #7:

- Definitions/Code:
    - Opening a text file. file = open("file_name.txt", "r") or w instead of r to write.

    - .close() = exits a file that was used for writing to properly save it.
        - Make sure to close both the read file and the write file.

    - Line = file.readline() = is used to read a line.
        - (Ex – this returns the value of the line followed by \n to denote the end of the line).
        - If the value in the txt file is an integer or float make sure to convert that after.

    - .write() = lets you write to the file specified. Print formats for strings work here.

    - raise Error ("Error message") = gives an error message, make sure to specify the type of error (IOError or ValueError for example). Usually put after an if.

    - try: and except Error as exception = use formatting like an if and else statement.

- finally: = is used to take an action regardless of whether or not the exception was raised. Used with try. NEVER USE finally: WITH except: .
  - However you can use (Ex – try:).

```
File = open("file.txt", 'w')
try:
    file.write("hello")
finlly:
    file.close()

except IOError:

    raise IOError ("file doesn't exist")
```

- with open|("file.txt", "w") as fileA: = is used instead of try/finally to pen a file and allow the user to modify the file, before it is automatically closed.

- Concepts:
  - Remember if you want to use backslashes double it as \\. Since \ is normally an escape character.

  - To separate a line into smaller segments one can use:
    - for line in file:
      line = line.split()
    - Use strip() or split(). You can also put and "r" or "l" in front of the strip/split to remove leading characters on that side.
      - This is most commonly used with .rstrip() to remove the \n on a returned string.
    - .strip() has further modifiers, (Ex – if a string = "a:bc:d" , then string.rsplit(":", 2) would give "a:bc" and "d" by splitting the string into 2 parts, with the split being made at the rightmost colon.

Lecture #8:

- Definitions/Code:
  - .set() = converts to a set (). Sets are unordered and have no reference-able positions. Sets are mutable. Set's faster then lists.
    - .add()
    - .discard()
    - .remove() = remove raises an exception if you try to remove a non existent element.

        - .clear() = removes all elements.
    - sorted() = returns a list (not a set) of the elements in an array.
    - .issubset() = returns True or False if a set is a subset of another set. A set is a subset if it has all the values that one bigger set has.
    - .union() = creates a new set combining 2 sets together with duplicates removed.
    - .intersection() = creates a new set of all common elements of sets combined.
    - .difference() = returns a new set containing elements of the first set that aren't in the second set.
    - .items() = returns a list of tuple pairs.
    - .get(key, backup) = returns the value associated with the key, and if the key doesn't exist the backup chosen value is returned. Backup not needed.
    - .values() = returns all values of the dictionary.

- Concepts:
    - Modules – with large files, you split the code up into separate source files called modules.
        - Driver modules – main().

Lecture #9:

- Definitions/Code:
    - def __init__(self): = a constructor.
    - self = used to reference the parent class of a method. Giving every object made by the class its own instance variables.
    - Mutators = setters.
    - Accessors = getters.

- Concepts:

- It is common practice in python to use "_" before a variable to show it is private.
  - (Ex - _item, which can be used as self._item = "").

- A method can have a default argument after the self, which will be used if the user does not input any parameters into the method.
- The only significant difference between a method and a function is that a method is able to operate on data within the class, whereas a function requires external input.

Lecture #10:

- Definitions/Code:
  - super(). = to reference a superclass constructor in place of self.
    - (Ex – def __init__(self, otherVariable):).
      super().__init__(otherVariable)

  - Polymorphism = the ability to use the same class reference for different forms. Where python can use any of the objects without needing to distinguish between the methods.
    - (Ex – referencing a method that is commonly named in multiple classes or subclasses).
