**DEMO CLASS 13th October 2017**

1. Introduction
2. Python Intro
3. Interactive Mode Programming
4. Python Identifiers (used to identify a variable , function, class, module or other object
   1. Reserved Words
   2. Lines and Indentation
   3. Multi-Line Statements
   4. Quotation in Python
   5. Comments in Python
5. raw\_input
6. Python Variable Types
   1. Int
   2. Float
   3. String
   4. Multiple assignment a = b = c = 1 Or a,b,c = 1,2,"john"
7. Standard Data Types
   1. Numbers
   2. String
   3. List
   4. Tuple *(The main differences between lists and tuples are: Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated. Tuples can be thought of as read-only lists.)*
   5. Dictionary (AKA: *associative arrays*)
8. Type casting(conversion):
   1. Ord():Converts a single character to its integer value.
   2. Hex():Converts an integer to a hexadecimal string.
   3. Oct():Converts an integer to an octal string.
9. Loops
   1. For
   2. While

**FIRST CLASS: 14th October 2017**

1. List intro and exercise
   1. Definition
   2. Indexing
   3. Slicing
   4. Repetition
   5. Concatenation
   6. append/deletion/pop/min/max
2. Tuple intro and exercise
   1. Definition
   2. Indexing
   3. Slicing
   4. Repetition
   5. Concatenation
   6. append/deletion/pop/min/max
3. Dictionary intro and exercise
   1. Definition
   2. Indexing
   3. Slicing
   4. Repetition
   5. Concatenation
   6. append/deletion/iterate over Items

**SECOND CLASS: 15th October 2017**

1. Making Decisions - if Statement
   1. If
   2. If..else
   3. Nested if..else
   4. Multiple Condition using logical operators
2. Loop Control
   1. While
   2. For
   3. Nested loops
   4. All loops with “else” statement
   5. Break , continue and PASS statement in loops

**Class Assignment**: multiple Each item of list1 with list2 elements.

* Store the result in new list.
* While performing this operation, filter even and odd numbers from result and store them in a separate list
* Make sure to remove duplicates from the resultant list

1. Iterator
   1. List iterator
   2. Tuple iterator
   3. Dictionary iterator

**All the above can be found here:**

**List & Loop:**

<https://docs.google.com/presentation/d/1x14FVuvcO7SNNW7VP3DbXXTmc6dIvnlMAtA8kQ-aR00/edit#slide=id.g1c1e2302a2_0_95>

<https://docs.google.com/presentation/d/12Pdc-AHCV3caTp2de9Nson4CI7AnxoEFB_VmZBNDrZU/edit#slide=id.g1c3cb20054_0_16>

**Tupple & Dictionary:**

<https://docs.google.com/presentation/d/1d8rCKdH5vqfrBx7vIcOz8kTMHYVuVx8sI8iEVUMGLXM/edit#slide=id.p>

**SECOND CLASS: 21st October 2017-No class**

**SECOND CLASS: 22nd October 2017-No class**

**THIRD CLASS: 28th October 2017**

1. **Functions & Scope of variables**
   1. Definition
   2. Syntax with example
   3. Calling function with reference and value
   4. Return statement
   5. Arguments:
      1. Required arguments
      2. Keyword arguments
      3. Default arguments
      4. Variable-length arguments
   6. The Anonymous lambda Function
   7. Use of filter,map and reduce with lambda function
   8. Scope of Variables local and global

**Exercise:**

* + - 1. Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters. Don’t use inbuilt functions.

**Homework:**

Write a Python program that accepts a hyphen-separated sequence of words as input and prints the words in a hyphen-separated sequence after sorting them alphabetically. Go to the editor Sample Items : green-red-yellow-black-white

Expected Result : black-green-red-white-yellow

Use reduce function and using “

+” operator, add only 1st element of passed number. Check the result?

E.g:

tmp = [10,2,3]

reduce(lambda x,y:x+1,tmp) **==??**

also

reduce(lambda x,y:x,tmp) **==??**

Using reduce function, retrieve only the 1st and last element of the given list

**FORTH CLASS: 29th October 2017**

**Modules**

1. What is module
2. Import statement
   1. Definition and its syntax
   2. Import single and multiple functions
   3. Import functions with aliasing
   4. From …Import statement
   5. From …Import \* statement
   6. Dir() function
   7. globals() and locals() functions
   8. reload() function

**Exercise:**

1. Write a module (with one line of documentation) to generate a list of reverse, even and odd numbers of a given list of integers
2. Call above written module using “**import”** in different file and load various available modules and perform the same operations (i.e. generate list of reverse, even and odd numbers)

**Homework:**

Continuation of class exercise:

**FIFTH CLASS: 04th November 2017**

**File I/O**

1. What is File
2. How to open File for following operations
   1. Read (r,r+,rb,rb+)
   2. Write (w,w+,wb,wb+)
   3. Append (a,a+,ab,ab+)
   4. Close
   5. File positions:
      1. tell()
      2. seek()
   6. next
   7. with statement
   8. file attributes (file.name/closed/mode)
   9. Some “os” related file operations
      1. rename
      2. remove
      3. mkdir
      4. chdir
      5. getcwd

**Exercise:**

1. Write a script which opens a text file and read the each line and print them along with line number.
2. Write a script which opens scripts itself in read only mode and print your program as output. Make sure that newline is not printed since each print statement will by default add one new line
3. Write a program to filter out those lines which is ending with even number in the statement. Also make sure that blank line is not available in final output.

Content of txt file:

This is line1

This is line2

This is line3

This is line4

This is line5

This is line6

This is line7

This is line8

This is line9

This is line10

**SIXTH CLASS: 05th November 2017**

**Logging**

1. What is Logging
2. How to log using “import logging” module

**Debugging**

1. What is debugging
2. How to debug your program using “import pdb” module
3. Frequently used and supported commands for pdb

**Exception**

1. What is Exception
2. How to catch exception using “Try … Except”
3. Catching cause of the exception using variable
4. Try … Except….Finally…. clause
5. Raise an exception using “raise …”
6. Assert an exceptione