**09-03-2019 :->**

Lec15 :->

Complex, Binary, Hexadecimal types

Lec17 :->

Type conversion function hex(), bin(), oct()

Lec19 :->

String repetition, substring

Lec21 :->

Steps in slicing, String reverse

Lec23 :->

String methods->>>>find(), count(), replace(), lower(), upper(), title()

**10-03-2019 :->**

lec\_24:->

#empty list

#list with data types

#indexing in list

#substring in list

#list repetition

#length of list

lec\_25 :->

#adding an element to the list

#deleting an element by value

#deleting an element by index value

Lec\_26 :->

#to remove all elements from list

#max and min element in the list (max() and min() function)

#insert element in the list at any position ------------- insert(postion,objcet)

#sort list

#sorting in reverse

**12-03-2019 :->**

Lec28

Use a tuple->>>> we cannot add or remove element from the tuple

Lec29

Conversion list to tuple

**12-03-2019 :->**

Lec30

Set->>>>> update (), remove ()

Lec30

Frozen set->>>>>> frozenset()

Lec32

Range function->>>>>> range(initial\_value, end\_value, steps)

Lec33

bytes() and bytearray()

lec34

dictionary ->>>>>>> keys(), values(),

**15-03-2019 :->**

Lec40

About print() method

Lec41

Print and string formatting

Lec43

Reading muiltiple input

**16-03-2019 :->**

Lec44

Student detail->> **program**

Lec45

Average of 3 numbers->> **program**

Lec46

Area of cicle

Lec48

Function as a argument to other function

**Control statements :->**

Lec51---**if else**

Even odd ->> **program**

Lec53—**if elif else**

Even odd handling zero->>**program**

Lec55—**while loop**

Display a number 1 to 10->>**program**

Lec56—**while loop**

Odd numbers between x and y->>**program**

Lec58—**for loop**

Display number from 50 to 70->>**program**

Lec59—**for loop**

Product of number In list->>**program**

Lec60—**for loop**

Table of given number->>**program**

Lec61—**break**

Break the loop if 5 is present->>**program**

Lec62—**continue**

Print 1 to 20 and skip multiple of 3->>**program**

Lec63—assert

Assert

**Command line argument :->**

Lec65

Display command line arguments and length

Lec66

Product of command line argument

**Function :->**

Lec70

Multiple return of function

Lec72

Accessing global variable with same name in a function

Lec73

Assigning a function to a variable

Lec74

Function inside another function

Lec75

Function can return a function

Lec76

Pass any data type to a function like list,tuple ,string , dictionary , set etc

**Recursion :->**

Lec78

Factorial of number with recusrsion

Lec79—Keyword argument

Keyword argument used with parameter to a function

Lec80

Default argument

**17-03-2019 :->**

**Lambdas:->**

Lec82

Cube of a number using lambda

Lec83

Even odd with lambda

Lec84

Sum of two numbers using lambda

Lec85—filter()

Getting even number from list using filter() function

Lec86—map()

Double the value of list

Le87—reduce()

Sum of list and highest in list

**Decorator :->**

Lec89—decorator

Defining a decorator function that double the value

Lec90-- @decorator

Defining a decorator function that double the value

Lec91--@decorator

Decorating a string

Lec93—generator are the function that produce the sequence of values

Create a generator

Generator are used with yield

**Module:->**

Lec94\_1 and Lec94\_2

Creation and use of module

Lec95-

Different ways of module

**List comprehension:->**

Lec97

Cube of numbers in the list

Lec97\_1

Set product of two list

A\*B={ ( a,b ) : a belong to A, b belong to B }

Lec97\_

Celsius to Fahrenheit ->> Converting Celsius list to Fahrenheit list

Lec98

Even numbers between 1 to 20 using list comprehension

Lec99

Product of two integer list

Lec100

Common element in two list

**Object Oriented Programming :->**

Lec104

Creating first class

Lec105

Parameterized constructor

Lec106

Defining the instance method

Lec107

Setter and Getter methods

Lec108

Defining instance method

Lec109

Define static field

Lec110

Define static method and count the number of objects

Lec111

Create a inner class

**19-03-2019🡪**

**Encapsulation**

Lec113-

Private Fields and name mangling

**Inheritance->**

Lec116

Inheritance implementation

Lec117

Inheriting the functionality

Lec118

Overriding

Lec119

Use of super()

**Polymorphism** 🡪

Lec121

Duck typing

Lec122

Duck typing for Dependency Injection

**Abstraction🡪**

Lec126

Creating an abstract class

Lec127

Creating Interface

**Exception Handling🡪**

Lec130

ZeroByDivision exception

Lec131

Finally block

Lec132

Else block

Lec133

Logging in action

Lec134

Logging configuration

Lec135

Implementing logging in program

Lec136

Using assertion

**File Handling🡪  
lec138**

Writing a string to a file

Lec139

Read a file

Lec140

Write multiple string

Lec141

Check if file exit or not

Lec142

Pickle

Lec143

Unpickle

**Regular Expression 🡪**

Lec145

\d 🡪 Digit 0-9

\D 🡪 Non Digit

\s 🡪 Whitespace

\S 🡪 Non whitespace

\w 🡪 alpha numeric it could be like qw23, gh5h etc.

\W 🡪 Non alpha numeric

\b 🡪 space around word

\A 🡪 matches at start of string

\Z 🡪 matches at end of string

Lec146

serach() 🡪It return the first matched word from the string and is used with group() method to display the result

lec147

findall() 🡪It will search all the matched word from string and return a list

match() 🡪It will search the only the starting word and return the object. With help of this object we call the group() method.

Lec148

Split()

Lec149

sub() 🡪 is used to substitute all the word in the string

lec151

quantifiers 🡪 +(all), \*(all), ?( zero or one character ), {m} , {m,n}

lec152

Finding dates

Lec154

^ 🡪to check first word

$ 🡪to check last word

**Date and time🡪**

Lec156

Time since epoch

Lec157

Finding the current date and time using datetime module

Lec158

Combining data and time

Lec159

Sorting the dates

Lec160

Sleep()

Lec161

perf\_counter()

21-03-2019🡪

**Threading🡪**

There are 3 ways of creating a thread.

Lec163

Main thread

Lec164

Thread using a function(The 1st way of creating thread)

Lec165

Printing thread name

Lec166

Thread extending the Thread class(The 2nd way of creating thread)

Lec167

Thread using a class (The 3rd way of creating thread)

Lec168

Multithreading in action

Lec169

Sleep()

Lec170

TicketBooking

Lec172

Add more logic to TicketBooking

Lec173

Synchronizing using lock()

Lec174

Synchronizing using semaphore()

22-03-2019->

Lec176

Thread communication using Boolean flag

Lec179

Thread communication using wait() and notify()

23-03-2019->

Lec181

Downloading HTML page

Lec182

Downloading Img

Lec184

Creating a server

**April2019->6-04-2019**

Collection\_module\_Counter.py

Collections\_module\_defaultdict.py

Collections\_module\_OrderedDict.py

Datetime\_module.py

Module\_pdb.py 🡪pdb=python debugger