**Exception Handling**

* used to simplify the error I.e Technical error has to be converted into user understandable format
* make the program to continue its execution
  + pre-defined exception handling
  + user-defined exception handling

**try:**

**{**

**statements**

**………….**

**………….**

**} ython3 exception4.py**

**except exceptionname:**

**{**

**errorhandling statement;**

**}**

* try can have more than one except statement

user-defined exception handling

1. **class userdefinedexceptionname(Exception):**

**pass**

2. when error occurs

**raise exceptionname**

**3.** writing of except statement to handle the errors

**Packages**

1. create a folder

2.under that folder create a python file

3.write the functions in that file

4. include the folder using the statements

f**rom foldername import pythonfilename**

**break and continue statement**

**1.break**

* used to terminate immediately

break

**2.continue**

* used to continue the loop statement for next iteration  
   continue

**Files**

**1.open()** --> opens the file for either writing or appending or reading

**filevariable=open(“filename”,”mode”)**

w-->write mode

r-->read mode

a-->append mode

**2.write()** --> used to write the single line of content into a file

filevariable.write(content)

**3.writelines()**--> used to write more than one lines at a time using list

filevariable.writelines(**list/listvariable**)

**4.read()**--> read char by chars

**5.readline()** --> read line by line

**6.readlines()**--> read all the lines from the file

**I.math-->** mathematical related functions are in this module

import math

Methods

1.factorial()--> returns the factorial of the value

5!=5\*4\*3\*2\*1

2.floor()--> returns the lowest value

3.ceil()--> returns the highest value

4.radians()--> converts degree into radians

5.degree()--> radians to degree

6.sin()--> returns the sign value

7.cos()--> returns the cos value

8.tan()--> returns the tan value

**II. random**

import random

generates random numbers

used in

Game Programming

Cryptography

Data Analysis

**Methods**

1.randint() --> generate the integer random numbers withinn the range

variable=random.randint(startvalue,endvalue)

2.random() --> 0 to 1 float random numbers excluding 1

variable=random.random()

3.randrange()--> generate random numbers within the specified range with step value

variable=random.randrange(startvalue,endvalue,stvalue)

4.uniform() -->generates random float numbers within the specified values

variable=random.uniform(startvalue,endvalue)

**III.OS Module**

import os

**1.system()**--> used to execute os commands

**os.system(“os commands”)**

2.os.access()--> checks the owners access permission of the file

os.access(“filename”,”Permission to be checked”)

os.R\_OK--> read permission

os.W\_OK --> write permission

os.X\_OK --> execute permission

os.F\_OK --> check for file existence

3. **os.walk()** --> list out the path,subdirectories and files of specified path

**var1,var2,var3=os.walk(path)**

**4.os.path.join()**--> joins the path and filename

variable=os.path.join(path,filename)

5.**os.path.getsize() -**-> returns the size of the file in bytes

variable=os.path.getsize(filename)

6.**os.path.getmtime()** -->returns the file modification time in floating value

variable=os.path.getmtime(filename)

7.os.getcwd()--> returns the current working directory

variable=os.getcwd()

8.listdir()--> lit the files and directories from the specified path

os.listdir(“path”)

9.mkdir()--> creates the new directory in the current path

os.mkdir(“directoryname”)

10.rename()--> changes the file or directory name

os.rename(“oldname”,”newname”)

11.remove()--> deletes the file

os.remove(“filename”)

12.rmdir()--> removes the directory if it empty

os.rmdir(“directoryname”)

13.rmtree()--> deletes the directory even it is not empty

shutil.rmtree(dirname)

14.chdir()-->change directory

os.chdir(“directoryname”)

**IV. Datetime,time**

1.ctime()--> converts the float format time into date time format

variable=time.ctime(floatvalue)

2.today()--> returns the current date

variable=datetime.date.today()

year-->separates the year

month--> separates the month

day-->separates the the date

3.now() --> returns the current date and time

variable=datetime.datetime.now()

4.strftime()-->format the date and time into string format

strftime(“attributes”)

%Y --> curent year

%B --> Month in text format

%W--> week of the year

%w -->weekday of the week in number

%j--> day of the year

%A--> day of the week in text

%d--> day of the month

5.timedelta()--> used to convert the nuymber into date format which helps to subtract and add the datetime

datetime.timedelta(days=number)

6.fromtimestamp()--> from the floating value it returns date and time in dateformat

datetime.datetime.fromtimestamp(floatvalue)

**psutil**

**matplotlib**

**mail sending**

**Boto3 package**

**API**

Example scripts