Day-7 Morning Assesment:

Exception Handling:

1. ‘except:’ is used when there is a imaginable exception raises like arithmetic exception, division error exception, etc… We can expect the exception and can handle the exception. ‘except Exception as e:’ is used when to catch the unimagined exception.
2. try:

num = input(“Please enter a number: “)

if not num.isdigit():

raise valueError(“Value Error Occured”)

else:

num = int(num)

if num == 0:

raise ZeroDivisionError(“Enter any number other than zero”)

else:

result = 100/num

print(result)

except valueError as ve:

print(“Enter numbers only”)

except ZeroDivisionError as ze:

print(“cannot divide any number with zero”)

1. Finally is a keyword which gives the user a conclusion of the process, it is normally used at the end of the program. Irrespective of the exception arose, finally is always executed.
2. class InvalidAgeError(Exception):

pass

try:

age = int(input(“Please enter your age: “))

if age < 18:

raise InvalidAgeError(“Invalid Age Error Occurred”)

else:

print(“You have a valid age”)

except InvalidAgeError as ie:

print(“You have an invalid age”)

1. Divided by Zero

Done

1. n=3

while(n>0):

try:

num = input(“Please enter a number: “)

if not num.isdigit():

raise ValueError(“Value Error Occured”)

n- = 1

else:

num = int(num)

print(100/num)

break

except ValueError as ve:

print(“Enter numbers only”)

else: print(“Failed 3 times”)



Regular Expressions

1. Pattern = r’^(?=.\*[A-Z])(?=.\*\d)(?=.\*[@#$%&]).{8,}$’
2. re.match() matches with given data. Matches only at the beginning of the string whereas re.search() search for the given data. Matches anywhere in the string.
3. import re

str = “Email me at [test123@gmail.com](mailto:test123@gmail.com) or [hr@openai.org](mailto:hr@openai.org)”

email = re.findall(r’[a-zA-z0-9.\_%+-][+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}’,text](mailto:+@[a-zA-Z0-9.-]+\.%5ba-zA-Z%5d%7b2,%7d’,text))

print(emails)

1. import re

num = input(“enter mobile number: “)

pattern = r’[6-9]\d{9}$’

if re.match(pattern, number):

print(“valid number”)

else:

print(“Invalid number”)

1. ^ - start of a string

[A-Za-z0-9\_] – Allows letters, numbers, underscore

{3,15} – length between 3 to 15

$ - end of the string

1. Text = “I love #Python and #MachineLearning! #AI”

Hashtags = re.findall(r’#\w+’, Text)

print(Hashtags)

1. re.match() used to check only the beginning of the string.
2. import re

def password():

password = input(“enter a password: “)

pattern = r’^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(/+.\*[@#$%&]).{8,}$’

if re.match(pattern,password):

print(“Valid Password”)

else:

print(“Invalid Password”)

password()