DAY-7 Morning Assessment

# EXCEPTION HANDLING

1.except exception as e:

Catches only exceptions derived from exception class and gives access to the exception object(e)

Except:

Catches all exceptions,including system-exit ones,which may lead to unpredictable behaviour or make the program hard to stop.

2. try:  
 num=int(input("Enter a number: "))  
 result=100/num  
 print(result)  
except ValueError:  
 print("error:Input must be a number.")  
except ZeroDivisionError:  
 print("error:You can't divide by zero.")  
except Exception as e:  
 print("Unexpected error:", e)

3.The finally block always executes whether an exceotion is raised or not. It is used for actions like closing files or database connections.

try:  
 file=open("file1.txt","r")  
 data=file.read()  
 print(data)  
except FileNotFoundError:  
 print("File not found")  
finally:  
 print("closing of file if opened")  
 if 'file' in locals() and not file.closed:  
 file.close()

4. class InvalidAgeError(Exception):  
 pass  
age = int(input("Enter your age: "))  
if age < 18:  
 raise InvalidAgeError  
else:  
 print("valid")

5. divided by zero

Done

6. for i in range(3):  
 try:  
 user\_input = int(input("Enter a number: "))  
 print("entered num:", user\_input)  
 break  
 except ValueError:  
 print("ValueError")  
else:  
 print("too many invalid attempts.exiting the program")

7.raise is used to manually trigger an exception.

Assert is used for debugging,checks if a condition is true.If not it raised an assertion error.

**def withdraw(amount):  
 if amount<0:  
 raise ValueError("amount cannot be negative")  
 print(amount)  
withdraw(-1)**

def withdraw(amount):  
 assert amount > 0,"amount must be greater than 0"  
withdraw(-1)

# REGULAR EXPRESSIONS

8. import re  
pattern=r"^(?=.\*[A-Z])(?=.\*\d)(?=.\*[@#$%&])"

9. import re  
text="Hello 123 World"  
print(re.match(r"\d+",text))  
print(re.search(r"\d+",text))

o/p: None

<re.Match object; span=(6, 9), match='123'>

**Re.match() checks only at the beginning of the string**

**Re.search() checks anywhere at the string.**

10. import re  
text="Email me at test123@gmail.com or hropenai@org"  
emails=re.findall(r"[a-zA-Z0-9.\_%+-]+@[a-zA-Z.\_]+\.[a-zA-Z]",text)  
print(emails)

o/p: ['test123@gmail.c']

11. import re  
def is\_valid(number):  
 return bool(re.match(r'^[6-9]\d{9}$',number))  
  
print(is\_valid('123456'))

12.This pattern matches:

Only strings that contained 3 to 15 characters

Allowed characters : A-Z,a-z,0-9 and \_

No special characters,no spaces

Entire string must match

13. text = "I love #Python and #MachineLearning! #AI"  
hashtags = re.findall(r"#\w+", text)  
print(hashtags)

o/p: ['#Python', '#MachineLearning', '#AI']

14.we can use re.compile to use it multiple times

import re  
pattern = re.compile(r"^\d{4}$")   
for val in ["1234", "abcd", "9876"]:  
   if pattern.match(val):  
       print(f"{val} is valid")

15. import re  
def is\_strong\_password(password):  
   pattern = r"^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)(?=.\*[@#$%&!^\*]).{8,}$"  
   return bool(re.match(pattern, password))  
user\_input = input("Enter your password: ")  
if is\_strong\_password(user\_input):  
   print("Strong password ")  
else:  
   print("Weak password ")