Date—18th June Assignment Exception and logs

1.What is the role of the 'else' block in a try-except statement? Provide an example scenario where it would be useful.

Ans—else block executes if try and exception block runs without error.

e.g. try:

n1 = int(input("Please enter n1: "))

n2 = int(input("Please enter n2: "))

result = n1 + n2

print("Result:", result)

except valueError:

print (“enter valid/ integer nos only)

else:

print(result) -- if we give float/string then except valueError will be executed

2. Can a try-except block be nested inside another try-except block? Explain with an

example.

Ans—yes we can nest try-except

e.g. try:

n1 = int(input (enter n1))

n2 = int(input(enter n2))

try:

result = n1 – n2

print(result)

except valueError:

print(“n2 > n1”)

except valueError

print (“error: enter integer numbers only”)

3. How can you create a custom exception class in Python? Provide an example that

demonstrates its usage.

Ans—class custSubError(Exception)

Pass

===================

def substract(a,b)

if a==b

raise custSubError (“same numbers not allowed”)

return a – b

====================================

4. What are some common exceptions that are built-in to Python?

Ans—common built in exceptions are – there are 30 to 40 built in errors in python ZeroDivisionError, ValueError, IOError, typeerror, keyerror, syntaxerror, indentation error etc.

5. What is logging in Python, and why is it important in software development?

Ans—When we run any code or do any action that event gets recorded in system this is called as log. E.g. if we are running a code or installing anything and suddenly we get an error in code like debug or critical error, then we come to know this when we check the logs of code.

6. Explain the purpose of log levels in Python logging and provide examples of when

each log level would be appropriate.

Ans—to understand the severity on any error we need to set logs

1 Debug—it’s a normal error and we can edit the code and debug the function

2 info—it gives the information about the code that is running

3. Warning—It euns the code but shows that if we don’t take any action on unexpected may happen.

4. ERROR: More serious problem that prevented the software from performing a function.

5. CRITICAL: A very serious error, indicating that the program itself may be unable to continue running.

7 What are log formatters in Python logging, and how can you customise the log

message format using formatters?

Log formatters means we set up the format of the log. E.g we can set the date, time and then message or we set the date time location (path—not much imp) and message

8. How can you set up logging to capture log messages from multiple modules or

classes in a Python application?

Ans—First we need to import the logging, then we mention the path and dir name and logfile name. then we join it so that we can find the saved logs in system easily.

9. What is the difference between the logging and print statements in Python? When

should you use logging over print statements in a real-world application?

-- Logs means the steps/ events that code is running, in print it will print only result or output of the function. Not an idea in real world how to use it

10. Write a Python program that logs a message to a file named "app.log" with the

following requirements:

● The log message should be "Hello, World!"

● The log level should be set to "INFO."

● The log file should append new log entries without overwriting previous ones.

-- no idea about writing the programme

11. Create a Python program that logs an error message to the console and a file named

"errors.log" if an exception occurs during the program's execution. The error

message should include the exception type and a timestamp.

-- no idea about writing the programme