**# 1. What is Thread and Multithreading?**

# Answer

# Thread: One independent path of execution through the code. The number of simultaneous active threads is up to the number of CPU’s or cores. Note that not all threads have to follow the same path, but the throughput increases if they do.

# Multithreading/multiprocessing: The ability to execute multiple processes simultaneously. If you have a color printer, it would be cumbersome if you need to print one color at a time, times the number of pages.

**# 2. What is Concurrency and Parallelism and what are the differences?**

# Answer:

# Concurrency is the task of running and managing the multiple computations at the same time. While parallelism is the task of running multiple computations simultaneously.

**# 3. What is Garbage collector? How does it work?**

**# 4. What is Transaction Management in a relational database (give an example)?**

# Answer:

# a Transaction is an entity – generally a group of SQL statements, or a combination of statements and stored procedure calls, which should be executed all or none.

# START TRANSACTION;

# SELECT @A:=SUM(salary) FROM table1 WHERE type=1;

# UPDATE table2 SET summary=@A WHERE type=1;

# COMMIT;

**# 5. What is an endpoint and what are the most common methods to interact withthe API data** source?

# Answer:

# a point at which an API -- the code that allows two software programs to communicate with each other -- connects with the software program.

# REST API

**# 6. What is data normalization in SQL? Please provide an example (any) of a database restructuring using primary/foreign keys to maintain data integrity.**

# Answer:

# Database Normalization is the process to eliminate data redundancies and store the data logically to make data management easier.

# FROM: EMPLOYEE ( < employee\_id >, name, social\_security, department\_name)

# To:EMPLOYEE\_DEPARTMENT ( < employee\_id >, department\_id )

# DEPARTMENT ( < department\_id >, department\_name )

**Section 2: Discuss Exception handing (4 pts) and debugging in Python (4 pts)**

Syntax:

try:

    # perform some task that may raise an exception

except Exception, value:

    # perform some exception handling

finally:

    # perform tasks that must always be completed

Exception only allows you to catch the exception you defined. Or the code with executed or stopped

The raise statement can be used to allow you to raise an exception where you deem appropriate. Using the raise statement, you can cause any of the Python exception types to be raised, you could raise your own exception that you define (discussed in the next section), or you could raise a string exception.

Debugging can be an easy task in Python via use of the assert statement

Assert synax:

assert expression [, message]

Assert expression cannot use list or triple

Assert function are commonly used in the test

**Q3: Write a function that takes in a non-empty array of integers that**

**are sorted in ascending order and returns a new array of the same**

**length with the squares of the original integers also sorted in** **ascending order.**

def new\_list(input\_list):

   new\_list= [num\*\*2 for num in input\_list]

   new\_list.sort()

   print (new\_list)

   return new\_list

numbers = [1,2,3,5,6,8,9]

new\_list(numbers)

**Q4 :Write tests for the newly created Sorted Squared Numbers function (in Q3). Provide a brief explanation for your test case options.**

#test

#there is repetitive number in the list:

numbers\_test1=[1,2,2,3]

assert new\_list(numbers\_test1)==[1,4,4,9]

# the order of input list changes

numbers\_test2=[3,2,1,6,9]

assert new\_list(numbers\_test2)==[1,4,9,36,81]

Q5 :**Agile methodology: name and describe any 2 of the main roles in a Scrum Agile team.**

ScrumMaster:  **a facilitator for an Agile development team. They are responsible for managing the exchange of information between team members.**

Product Owner :  **Product Owners manage the product backlog and ensure the company gains maximum value from the product.**

**Q6: Discuss advantages and disadvantages of TDD (Test Driven Development):**

**Pro:**

* **The code is effective:** you’ve got to prevent writing production code when all of your tests pass. If your project needs another feature, you would like a test to drive the implementation of the feature. The code you write is the simplest code possible.
* **Easier for unitest:** The code was developed based on each module. Therefore complex request will be reduced easy for debugging
* **Con:**

Additional work slow the process

Additional work when the requirements and feature get changes

**Q7: What is a Python DB cursor? Provide an example**

A cursor is an object which helps to execute the query and fetch the records from the database. The cursor plays a very important role in executing the query. This article will learn some deep information about the execute methods and how to use those methods in python.

#establishing the connection

import mysql.connector

conn = mysql.connector.connect(

user='root', password='password', host='127.0.0.1', database='mydb'

)

#Creating a cursor object using the cursor() method

cursor = conn.cursor()

8. Given an example table below

• Write a SQL query to find the maximum order (purchase) amount for

each customer.

• The customer ID should be in the range 3002 and 3007 (begin and

end values are included.).

• Filter the rows for maximum order (purchase) amount is higher than

1000.

• Return customer id and maximum purchase amount.

USE OrdersInfo;

select a.customer\_id, max(a.indivudal\_purchase) from

(select customer\_id,max(purch\_amt) as indivudal\_purchase from ORDERS

where customer\_id between 3002 and 3007

group by customer\_id having max(purch\_amt)<1000) a

9. TWO NUMBER SUM: ● Write a function that takes in a non-empty array of distinct integers and an integer representing a target sum. If any two numbers in the input array sum up to the target sum, the function should return them in an array, in any order. If no to numbers sum up to the target sum, the function should return an empty array. ● Note that the target sum has to be obtained by summing two different integers in the array. You cannot add a single integer to itself in order to obtain the target sum. ● You can assume that there will be at most one pair of numbers summing up to the target sum. Sample Input: numbers = [3, 5, -4 ,8, 11, 1, -1, 6] target\_sum = 10 Sample Output: [-1, 11] the numbers can be in any order, it does no

def sumtest(list,target\_value):

    output\_list=[]

    i=0

    num=1

    while i < len(list)-1:

       print(f'i={i}')

       while num<len(list)-1:

        print (num)

        print(list[i]+list[num])

        if list[i]+list[num]==target\_value:

          output\_list.append(list[i])

          output\_list.append(list[num])

          break

        num+=1

       if len(output\_list)==2:

        break

       else:

        i+=1

    print(output\_list)

    return output\_list