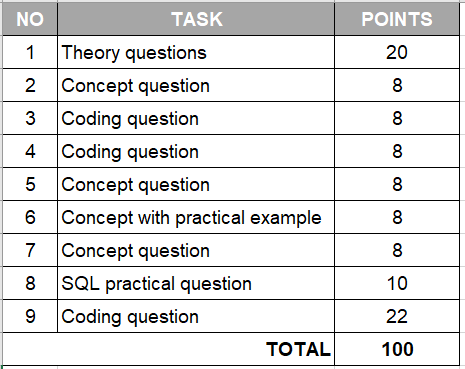
**ASSESSMENT**

Python and MySQL

assessment test 2 hours



|  |  |
| --- | --- |
| 1. **Python theory questions** | **20 points** |

1. What is the program?

A set of instructions that are run by a computer to perform tasks.

1. What is the process?

An action performed by a computer when it receives instructions.

1. What is Cache?

A small amount of memory that is part of the CPU. It temporarily holds instructions and data that needs to be accessed quickly and used often.

1. What is Thread and Multithreading?

A thread is a single process of a command. Multithreading is when multiple commands are executed at the same time.

1. What is GIL in Python and how does it work?

GIL stands for Global Interpreter Lock. It manages threads so that only one is executed at a time. This prevents data from being accidentally overwritten by multiple threads trying to access the same memory location.

1. What is Concurrency and Parallelism and what are the differences?

Concurrency – running and managing different computations at once

Parallelism – running multiple computations simultaneously

1. What do these stand for in programming: DRY, KISS, BDUF

Don’t Repeat Yourself, Keep It Simple Stupid, Big Design Up Front

1. What is Garbage collector? How does it work?

It is a built-in feature in python where it regularly deletes unused objects to free-up space in memory.

1. What are ‘deadlock’ and ‘livelock’ in a relational database?

Deadlock – everything stops because two or more transactions are waiting for the other to give up locks. One transaction must be aborted before activity starts again.

Livelock – requests for an exclusive lock are repeatedly denied because of overlapping shared locks that keep changing the status and prevent the task from being completed.

1. What is Flask and what can we use it for?

Flask is a web framework that can be used to build a web application.

|  |  |
| --- | --- |
| 1. **Discuss the difference between Python 2 and Python 3** | **8 points** |

* The print keyword in python 2 was replaced with the print() function in python 3
* When the division operator is used in integer division in python 2, the output is an integer even if the answer is a decimal number, whereas python 3 returns a float if the answer is a decimal number
* The xrange() function in python 2 is replaced by the range() function in python 3
* Python 3 uses Unicode strings by default whereas Python 2 stores strings as ASCII by default

|  |  |
| --- | --- |
| 1. **Write a function that can define whether a word is a Palindrome or not (a word, phrase, or sequence that reads the same backwards as forwards, e.g. *madam*)** | **8 points** |

def isAPalindrome(string):

return f'Is {string} a palindrome? {string == string[::-1]}'

print(isAPalindrome(input("Please enter a word to see if it is a palindrome: ")))

|  |  |
| --- | --- |
| 1. **Write tests for the newly created Palindrome function. Provide a brief explanation for your test case options.** | **8 points** |

import unittest

from palindrome\_program import isAPalindrome

class TestPalindromePrograme(unittest, TestCase):

def test\_ isAPalindrome(self):

expected = 'Is madam a palindrome?’ True

result = isAPalindrome(‘madam’)

self.assertEqual(expected, result)

if \_\_name == ‘\_\_main\_\_’:

unittest.main()

|  |  |
| --- | --- |
| 1. **Agile methodology, Scrum: name at least 3 types of meetings that are exercised by Agile teams and describe the objective of each meeting.** | **8 points** |

Sprint planning – the scrum team meet to plan each week of their sprint. Tasks are assigned and deadlines are set.

Sprint review – the team demonstrate and present their work to the product owner and stakeholders to gather feedback on the features and functionality of the product.

Sprint retrospective – meeting held at the end of each sprint for the scrum team to discuss what went well, what didn’t go well, and what improvements can be made in preparation for the next sprint.

|  |  |
| --- | --- |
| 1. **Exception handling in Python, explain what each of the following blocks means in the program flow:**   Try, except, else, finally | **8 points** |

Try – tests the code where the error is expected / anticipated

Except – allows you to handle the error if it happens

Else – this code is executed if there is no exception

Finally – always gets executed whether or not there is an exception

|  |  |
| --- | --- |
| 1. **How can we connect a Python program (process) with a database? Explain how it works and how do we fetch / insert data into DB tables from a python program.** | **8 points** |

You pip install mysql in the python editor then add host, user and password to the config.py file.

You connect to the database through the mysql connector then use a function in python to pass in the host, user and password to access the database.

To fetch or insert data into the DB tables, you have to write the SQL code inside the python functions. The cursor is used to fetch each row, the SQL query is executed, then changes are committed to the database.

|  |  |
| --- | --- |
| 1. **Given two SQL tables below: authors and books.**  * **The authors dataset has 1M+ rows** * **The books dataset also has 1M+ rows**   Create an SQL query that shows the TOP 3 authors who sold the most books in total! | **10 points** |

SELECT DISTINCT (a.author\_name), COUNT(b.sold\_copies)

FROM authors a

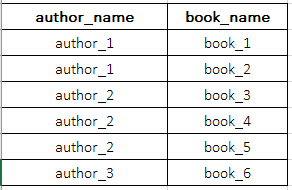
INNER JOIN books b

ON a.book\_name = b.book\_name

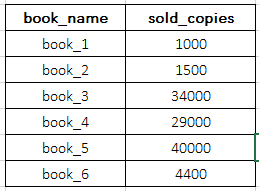
ORDER BY DESC

LIMIT 3;

**AUTHORS**

****

**BOOKS**

****

|  |  |
| --- | --- |
| 1. **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 not matter. | **22 points** |

my\_numbers = [3, 5, -4, 8, 11, 1, -1, 6]

target\_sum = 10

""" Takes the sorted list and creates a pointer for the end value.

Iterates through the sorted list and checks:

a. if sorted\_list[i] and end\_pointer are at the same index => skips to re-assigment of end\_pointer to next index in

b. checks if sorted\_list[i] + end\_pointer equals target\_sum => returns [value1, value2]

If the target\_sum cannot be reached by adding any two values in the list => returns [] """

def twoNumSum(sorted\_list):

end\_pointer = sorted\_list[len(sorted\_list)-1]

for j in range(len(sorted\_list)):

for i in range(len(sorted\_list)):

if sorted\_list[i] == end\_pointer:

continue

elif sorted\_list[i] + end\_pointer == target\_sum:

return [sorted\_list[i], end\_pointer]

end\_pointer = sorted\_list[len(sorted\_list)-j-1]

if sorted\_list[i] + end\_pointer != target\_sum:

return []

# Sorts the list from lowest to highest number and returns sorted list

def sortList(list):

list.sort()

return(list)

print(twoNumSum(sortList(my\_numbers)))