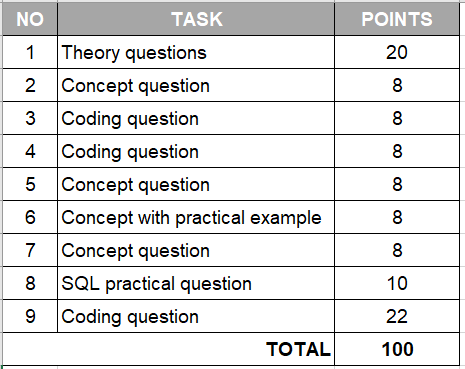
**ASSESSMENT**

Python and MySQL

assessment test 2 hours

Michaela D’Mello



|  |  |
| --- | --- |
| 1. **Python theory questions** | 1. **oints** |

1. **What is the program?**

Instructions a computer executes to perform a task.

1. **What is the process?**

The code and activity of a program which can be executed by one or multiple threads.

1. **What is Cache?**

A temporary small amount of memory within CPU that the CPU is likely to reuse

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

Thread = one command at a time

Multithread = processes multiple commands at once

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

GIL = Global Interpreter Lock

Effectively makes any CPU-bound Python program single-thread

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

Concurrency = more than 2 tasks can occur at overlapping times, but not necessarily at the same time

Parallelism = tasks can occur at literally the same time

Example using retail: 2 lines for 1 cashier = concurrency vs. 2 lines for 2 cashiers = parallelism

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

DRY= don’t repeat yourself

KISS= keep it simple stupid

BDUF= Big Design Up Front

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

Garbage collector = automatic process to free up memory when not in use to make way for other, new objects

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

Deadlock = 2+ tasks waiting indefinitely for another task to release a lock (...?)

Livelock = 1 task has to wait indefinitely to acquire a lock

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

Flask is a Python web framework that is used to develop web applications.

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

Python 3 is easier to understand than Python 2 as it has simpler syntax and more intuitive. For example, Python 3 uses Unicode, whereas Python 2 uses ASCII, and will return a decimal rather than whole number for divisions.

It also has more libraries, so Python 3 is more versatile than Python 2.

Range() in Python 3 has replaced rangex from Python 2.

|  |  |
| --- | --- |
| 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** |

# Create a function that returns the reverse of word, which matches the original string

def is\_Palindrome(s):

   return s == s[::-1]

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

import unittest

from unittest import TestCase, main

class Test\_Palindromes(TestCase):

   def test\_correct(self):

       s = "madam"

       ans = is\_Palindrome(s)

       if ans:

           print("Yes")

       else:

           print("No")

       self.assertTrue(ans)

   def test\_incorrect(self):

       s = "banana"

       ans = is\_Palindrome(s)

       if ans:

           print("Yes")

       else:

           print("No")

       self.assertFalse(ans)

if \_\_name\_\_ == '\_\_main\_\_':

   main()

First test tests a correct palindrome in which a ‘Yes’ should be printed, whereas the second test tests an incorrect palindrome in which a ‘No’ should be printed.

|  |  |
| --- | --- |
| 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** |

1. Daily Scrum
   1. Generally, no longer than 15 minutes
   2. Facilitated by Scrum Master
   3. Each person answers: what they did yesterday, what they are doing today, whether there are any obstacles they are facing
2. Sprint Planning
   1. Should take no longer than 6 hours
   2. Whole team discusses estimated effort involved for each team
   3. User stories are assigned to individuals
3. Sprint Retrospective
   1. Following the end of the whole project
   2. Should be no longer than an hour
   3. Includes Scrum Master, Product Owner and Dev Team
   4. Place to highlight what worked, what didn’t and what they can learn for similar projects in the future

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

A “try” clause within a Python program works along with the “except” clause. The “try” clause is always executed until the “except” clause is encountered.

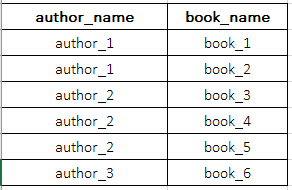
“Else” usually used in conjunction with an “if” statement, it is run when the previous command is not true.

“Finally” is a final step that will always run, similarly to the “try” clause. Whether the code runs the except or else clauses or not, the “finally” clause will always run.

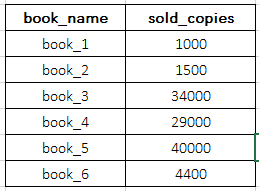
|  |  |
| --- | --- |
| 1. **How can we connect a Python program (process) with a database? Explain how it works and how do we fetch / insert data into DB tables from a python program.** | **8 points** |

|  |  |
| --- | --- |
| 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** |

**AUTHORS**

****

**BOOKS**

****

SELECT book\_name, sold\_copies FROM Books  
INNER JOIN AuthorsON Authors.book\_name = Books.book\_name

ORDER BY sold\_copies;

|  |  |
| --- | --- |
| 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** |

# Create a function

# Identify 2 integers from the list

# Ensure the 2 integers are not the same integer

# The two integers should equal the target

# Don't know how to return an empty array if no 2 numbers sum up to the target sum

# Print all options as a list

def my\_function(numbers):

   for x in numbers:

       for y in numbers:

           if x != y and x + y == target:

               print([x, y])

# Define list and target sum

numbers = [3, 5, -4, 8, 11, 1, -1, 6]

target = 10

# Call function

my\_function(numbers)

#Output

[11, -1]

[-1, 11]