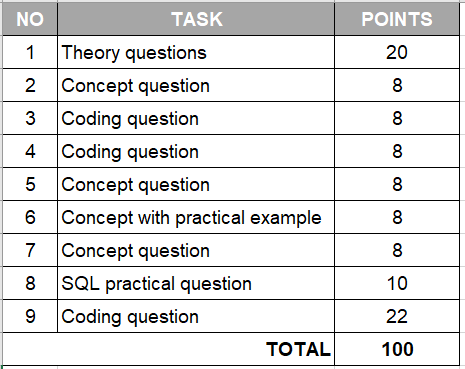
**ASSESSMENT**

Python and MySQL

assessment test 2 hours



|  |  |
| --- | --- |
| 1. **Python theory questions** | **10 points** |

1. What is the program?

A program is a set of instructions that a computer runs, it is used to build and run algorithms.

1. What is the process?

A process is an executed program, and many processes can be run at the same time in the background by the computer.

1. What is Cache?

A cache is a stored set of instructions by the CPU in the RAM to allow for easy and quicker retrieval of them in the future.

1. What is Thread and Multithreading?

A thread can be created when building programs and will run instructions and processes in and is used to be able to run ‘together’ at the ‘same time’, where the threads are executed so fast between each other that it seems simultaneous, which is what is referred to as multithreading, an example could be a thread that takes case of the GUI of a program while another thread will be built for the backend processing of the data to be fed back to the user.

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

The GIL, is the global interpreter lock and its main purpose is to be able to lock the interpreter to whoever is holding the lock, i.e other threads will not be able to execute with the interpreter as only the one holding lock can use it.

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

**Concurrency** is when multiple processes are run in what seems to be at the same time, running so fast concurrently that it seems to be running together, whereas **Parallelism** refers to executions on the same data in parallel to each other.

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

**DRY**: don’t repeat yourself, refers to not repeating the same code over and over again, but rather modularising code bites in order for it to be reusable.

**KISS**: keep it simple stupid, refers to not overly complicating the code, making sure to keep code as simple as possible, and clear naming where possible and documenting as you code in order to keep it simple and readable for yourself and for future readers of the code.

**BDUF**: big design U? first: this concept refers to creating the system design first and formost before even thinking about coding it up. The whole system is planned, designed and documented which will make it much easier and hopefully bug free and not error prone when coming to coding it all up.

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

The garbage collector is used by python to manage memory when coding. it works by collecting any unreferenced and lost variables within out programs when executed, and thereby freeing up that memory space to be used by something else. Variables can be lost if the reference pointer changes to a different to new variable, thereby allowing that memory space to be collected by the garbage collector and freeing up memory.

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

Deadlock and livelock are situations that can occur when multiple transactions are trying to use the same resources within a database. Deadlock refers to when one transaction is holding a resource that a newer transactions needs, that is also holding another resource that the older transaction needs, for example. As both transactions are in need of some resource, they are essentially waiting until they ‘die’. The database management system is built in order to handle these situations, by either waiting it out, or killing on transaction. This means that it will allow the transaction to wait a little longer and allow one transaction to carry out its task, or it may decide to kill one transaction for a fraction of a second, then run it again once the resources it needs have been freed and then run it again with the same timestamp to make it seem simultaneous.

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

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

Flask is a framework that holds various methods and pre-built components and codes, that make it much easier to easily and quickly build apps, e.g. it can build websites easily by creating ‘routes’ as the links to pages, therefore allowing for a much faster build time, that is also well established and widely used in industry and so is recognisable and very readable. It is also known to be very beginner friendly.

There are many differences between python 2 and python 3. Here are four examples:

**Stability:** python 2 is considered a more stable version as it has been out for longer and has a much richer library compared to python 3.

**Division:** one difference is in how the division works between the two versions. 5/2 in python 2 would return an int, 2; whereas 5/2 in python 3 would return a float, 2.5

**Printing:** a third difference is in python 2 the print is a statement, e.g. print “hello world”, whereas in python 3 it is a function, e.g. print(“hello world”)

**Strings:** one difference is that python2 had two different ways of representing strings, whereas in python3 all strings are now in Unicode.

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

*In Python answer file*

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

*In Python answer file*

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

**Standup meeting:** is held usually on the same time and place in order for the development team to discuss what they’re working on, if anything is blocking their work or they are stuck on anything, and to get some help or share thoughts and opinions, this allows for quick thinking and quick change of plans if there needs to be that all fit along the ‘agile’ principles of development to get the product out

**Planning meeting:** this is the initial planning stage meeting where the scrum master, and other key stakeholders will provide key requirements and specifications for the project, for the scrum master to present to the team and they will all plan the next sprint together with as much planning and detail as possible in order to meet the requirements and make the tasks as doable as possible while estimating the time needed and manpower estimated for the tasks in order to establish time frames for the next sprint until the final product is able to be delivered, as either a final product or MVP, etc.

**Reflection meeting:** is the final meeting in which they team will come together and discuss what went well in the sprint, what they could improve on, with some actionable feedback, and it’s a place to demo the product.

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

**Try:** this is where all the code that you want executed but you know some parts of it might raise an error will sit. We place conditionals in order to catch the exception raising circumstances, and within it raise the exception.

**Except:** this is where any raised exception in the try block will come to, and any code within this block will now be executed, it can print out the exception or print some information for the developer to understand that an exception was raised to due to whatever specified and explained circumstance

**Else:** this is where the rest of the code to be executed will site if the try is successful, and so it will be run, however this block of code will not run if an exception is raised while in the try block that takes us into the except block

**Finally:** this will run no matter what happens in the try block, if an exception is raised and a printed message is printed to console in the except block, then the next piece of code to run will still be the finally block, which again will run after the else block if the try block was successful.

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

We use the mysql.connector library that we import into our python file. We also need to make sure we have a separate file where we hold the access details for the database, specifically the username, the password and the URL, e.g. in our case it was local host. Once we have those details stored in a separate file we need to import that into our main connector python file. Now first we need to start a connection, which we previously called ‘cxn’, as a variable and initialise a connection object, and open a connection by using the mysql.connector.connect() function and pass into it the username, and password and URL we previously had before. After starting this connection, we now need a cursor that will be able to run our queries. This is essentially a variable, we used ‘cur’, and we create it by writing cxn.cursor(). Once we have done that we can create query variables by writing our queries in strings, typically held within triple quotes, and then we pass that query variable into the cursors execute method, e.g. cur.execute(query). Once executed we then call the fetchall() method on the cursor we built and save its returned contents into a variable, e.g. results = cur.fetchall(). This returned variable is iterable, and so we can iterate through our ‘results’ variable and see the values returned from our query. So that is how we can fetch data from our database. On the other hand, we can write our query to update or to insert, and we can format the string to accept variables that can be inserted into the database. We need to make sure to close the cursor once we have executed a query and fetched the results, and then make sure to close the connection if the is still connected to the database, both can be done using the .close() method.

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

Table

Description automatically generated**Table

Description automatically generated**

SELECT DISTINCT a.author\_name, SUM(b.sold\_copies) as total\_sales

FROM authors as a

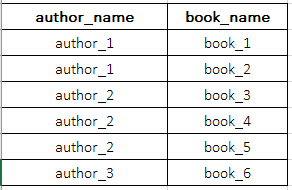
JOIN books as b

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

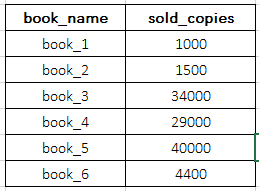
GROUP BY a.author\_name

ORDER BY total\_sales DESC;

**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** |

*In TwoNuberSumAnswer Python file*