

Assessment 2

1.

1.1 What is the program?

The program the name given to a set of code that is executed to perform a specific task.

1.2.What is the process?

The process is the term used when a programme is being executed

1.3.What is Cache?

The cache is a type of memory. When running a program, a copy of the program is stored in the cache

1.4.What is Thread and Multithreading?

Thread refers to the smallest piece of executable code. Multi-threading is when many threads are ran at once.

1.5 What is GIL in Python and how does it work?

So, annoying I forgot!

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

Concurrency is the process of running 2 programs at the same time. Parallelism involves running the 2 programs serially (i.e., one after another). Parallelism would be necessary if the input of one program is dependent on the output of another program, meaning that if it was executed it would not yet have the input. Concurrency allows for multi-tasking on the user's part.

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

DRY = don't repeat yourself (e.g., make sure variables are only defined once)

KISS = keep it simple stupid! (i.e., the aim is for the code to make sense to others)

BDUF = can't remember the acronym! But it's purpose is to promote good practice in coding language

8.What is Garbage collector?

The garbage collector is an in-built feature of python. It stores the names of all object variables/values, and looks for instances of these in scripts. If it finds that the objects do not exist anywhere except in the garbage collector anymore, then it is removed in order to free up memory.

9.What are 'deadlock' and 'livelock' in a relational database?

Both refer to the inability to execute queries.

A deadlock occurs when queries cannot be executed as it is waiting for some other process to have occurred.

A livelock is like too much is trying to happen at once, meaning that no one thing can happen in a given moment – something has to give.

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

Flask is a library. When imported into Python, it can help to build application programming interfaces.

2.

Python 3 is the newest version of Python, and Python 2 is the older. There are a number of key differences between these two versions.

- The print statement differs, python 3: print(), python 2: xprint()
- Python 3 is compatible with a larger number of libraries
- Python 3 is more high-level and so the interpreter often knows what datatype it is dealing with without the programmer having to explicitly state
- Python 3 is faster
- Memory is managed differently in python 2 and 3

3.

```
#get input
word = input("Enter a word:")
#define function
def palindrome(word):
    #for each letter
    for letter in word:
        #get the index of the letter in a number format
        i = letter.index()
        #if the letter is equal to the letter that occurs when you negatively slice the
        string
        if word[-i] == letter:
            #say it is a palindrome
            return "is palindrome"
            #else, say it isn't
        else:
            return "is not palindrome"
```

4.

#this code assumes that the function in Q3 is correct (which I'm sure it actually isn't!!)

```
#import testing pre-requisites
from unittest import TestCase, main
from file_where_palindrome_is import palindrome
```

```
#write necessary syntax for testing
class TestPalindromeCorrect(self):
    #define expected and result
    expected = 'is palindrome'
    result = palindrome(word='Hannah')
    #assert equal
    self.assertEqual(expected,result)
    return
```

```
#copy of above but testing for non-pallindrome input
class TestPalindromeIncorrect(self):
    #define expected and result
    expected = 'is not palindrome'
    result = palindrome(word='Charlotte')
    #assert equal
    self.assertEqual(expected,result)
    return
```

5.

Daily scrum – this is a 15-minute meeting at the beginning of each day of a sprint. Scrum team members answer 3 questions: what did they do yesterday? what are they doing today? Do they have any blockers/impediments to their work?

Scrum retrospective – this is done after a sprint has finished. It is chance for Scrum team members to reflect of the sprint; the highs, the lows, and where they could have improved. The point is to improve the quality of sprints in the future, by incorporating the feedback from this meeting into future sprint planning.

CAN'T REMEMBER THE NAME BUT IT BEGINS WITH AN 'R' I THINK? – this is done after a sprint, and is a chance for Scrum members to showcase the output of their work during the sprint to other team members.

Sprint planning – this is a meeting where the Scrum team go through the backlog of items (i.e., a list of potential things to improve the product; created from feedback from users) and after prioritising the items, segment them in tasks that can be completed by an individual(s) during a sprint.

6.

Try – tells python to attempt to execute the code in the block

Except – tells python that if an exception (i.e., an error) arises in the ‘try’ block, then it should execute the code in this block instead of raising the default, in-built error. The exception might simply be a nicer way of writing the error using a print statement, or it might contain code that deals with the exception in a different way.

Else – tells python that if it cannot carry out the exception block – i.e., if the reason that the ‘try’ block failed was not because of the expected exception, then it should execute this alternative code

Finally – no matter which of the above blocks are executed, python should always end by running this block

7.

Step 1: make a python file that contains the configuration information for the database. E.g., if the database is in MySQL, then the python file should store the username, host name, and password credentials for MySQL in separate variables

Step 2: create a python file that connects python and the database. The file must import the credentials made in step 1, as well as libraries such as MySQL. It must then create a function, using modules such as MySQL.connector, which connects python with the database.

Step 3: once the connection has been established in step 2, the same python file used in step 2 can be used to query the data in the database. To do this, a function is made that outlines exactly what you want to do (e.g., fetch data).

Step 4: to actually run this program, a ‘main method’ needs to be defined at the end of the script.

8.

-- use inner join to link databases

```
SELECT *
```

```
FROM authors a
```

```
JOIN books b
```

```
    ON a.book_name = b.book_name
```

```
ORDER BY sold_copies DESC
```

```
LIMIT 3
```

-- sorry, I know this isn't in total!! I couldn't quite figure it out though. Something along the lines of GROUP BY and use author_name...?

9.

```
#define function
```

```
def calc(my_numbers, target_sum):
    length = len(my_numbers)
    null_res = []
    output = []
    # for each number
    for num1 in range(length - 1):
        # add that number, and the number with all the other numbers
        for num2 in range(num1 + 1, length):
            # if the sum of the 2 numbers equals the target sum
            if my_numbers[num1] + my_numbers[num2] == target_sum:
                # put the 2 numbers in a list
                output.append(my_numbers[num1])
                output.append(my_numbers[num2])
                #and print it!
                print(output)
                #and stop the loop as soon as you find one!
                pass
            #if they don't add up
        else:
            #return an empty list
            print(null_res)
```

```
#example of using the call function
calc([7,8,9,10], 12)
```