

# THEORY QUESTIONS ASSIGNMENT

## Python based theory

To be completed at student's own pace and submitted before given deadline

### 1. Python theory questions

**What is Python and what are its main features?**

Python is a high-level, object oriented, programming language.

Its free source and dynamic

**Discuss the difference between Python2 and Python3**

Python 3 has a better syntax and it handles variables better, global variables do not change when use in a loop. It also includes multiple libraries that are not av available for python 2

**What is PEP 8?**

**It provides guidelines and best practices for coding in python**

**In computing/computer science what is a program?**

**A program is a set of instructions in a programming language that the computer follows**

**In computing / computer science what is a process?**

**Is an action being executed**

**In computing / computer science what is cache?**

**Is used to temporarily store data**

**In computing / computer science what is a thread and what do we mean by multithreading?**

**A thread is a self contained sequences of instructions, and multithreading allows for 2 or more threads to execute at the same time**

**In computing / computer science what is concurrency and parallelism and what are the differences?**

**Concurrency is when two or more tasks can start, run, and complete in overlapping time periods. Parallelism is when tasks literally run at the same time, e.g., on a multicore processor.**

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

**The GIL is a single lock on the interpreter itself which adds a rule that requires acquiring the interpreter lock to execute. It essentially makes the program single-threaded.**

**What do these software development principles mean: DRY, KISS, BDUF**

**DRY - Dont repeat yourself . Instead of repeating functions, call them**

**KISS - Keep it simple stupid. Keeping code as simple as possible and avoid overcomplicating things**

**BDUF - Big Design Up Front. Start small and reiterating**

**What is a Garbage Collector in Python and how does it work?**

**Keeps track of all objects in memory and it works by deleting the ones not in used**

**How is memory managed in Python?**

**Python involves a private heap containing all Python objects and data structures and is ensured internally by the memory manager.**

**What is a Python module?**

**A file containing statements and definitions**

**What is docstring in Python?**

**Is a string used to document and explain what a module, class, function or method do**

**What is pickling and unpickling in Python? Example usage.**

**pickling is a process by which the object structure in Python is serialized. A Python object is converted into a byte stream when it undergoes pickling. Unpickling is a process by which original Python objects are retrieved from from the pickle file.**

```
import pickle
l = ['a', 'b', 'c', 'd']
with open('file.txt', 'wb') as f:
    pickle.dump(l, f)
#UNPICKLING
```

```
import pickle
pickle_off = open ("file.txt", "rb")
ex = pickle.load(pickle_off)
print(ex)
```

**What are the tools that help to find bugs or perform the static analysis?**

**How are arguments passed in Python by value or by reference? Give an example.**

Using calling. parameters (arguments) in the Python language are passed by reference. If you pass immutable arguments like integers, strings or tuples to a function, the passing acts like Call-by-value.

```
def count(list)//by reference
c = 0
for i in list:
c += 1
fruits = apples, kiwis, pineapple, strawberry, blueberry
count(fruits)//by value
```

**What are Dictionary and List comprehensions in Python? Provide examples.**

**What is namespace in Python?**

Collection of names with information about the object that each name references

**What is pass in Python?**

Placeholder for future code, but to continue without errors

What is unit test in Python?

Checks every component of the code

In Python what is slicing?

It allows to access certain parts of a list, tuple or strings

What is a negative index in Python?

Its used to index from the end

How can the ternary operators be used in python? Give an example.

They can be used as a conditional, if the statements marks true or false

```
Max = a if a > b else b
```

What does this mean: \*args, \*\*kwargs? And why would we use it?

\*args is used to pass a non-key worded, variable-length argument list

\*\*kwargs used to pass a keyworded, variable-length argument list

How are range and xrange different from one another?

range returns a Python list object and xrange returns an xrange object

What is Flask and what can we use it for?

framework that provides useful tools and features that make creating web applications in Python easier

What are clustered and non-clustered index in a relational database?

Clustered is to define the order or sort the table all together

Non clustered keeps the data at one place and the record at anoter

**What is a 'deadlock' a relational database?**

**Two or more transactions are waiting for one another to give up locks**

**What is a 'livelock' a relational database?**

**request for exclusive lock is denied continuously because a series of overlapping shared locks**

## 2. Python string methods:

**Describe each method and provide an example**

- **capitalize()**
  - It capitalizes every word

```
str = "we like to capitalize each word"  
print(capitalize(str))
```

- **casefold()**
  - Turns all characters into lower case

```
str = "WE LOVE LOWERCASE"  
print(casefold(str))
```

- **center()**
  - Aligns the string to the center

```
str = "to the center"  
print(center(str))
```

- **count()**
  - Counts the number of instances

```
numbers = [1, 4, 2, 9, 7, 8, 9, 3, 1]  
print(numbers.count(9))
```

- **endswith()**
  - Checks if a string ends with the argument
  - Returns true or false

```
str = "What is up?"  
print(str.endswith("?"))
```

- **find()**
  - Find the first instance of the argument
  - Returns the index

```
str = "where is the h?"  
print(str.find("h"))
```

- **format()**
  - Formats a string by placing the value in the place holder

```
str = "its {price:.2f} euros!"  
print(str.format(price = 20))
```

- **index()**
  - Returns the index of the element in the list

```
l = ["apple", "kiwi", "strawberry", "grapes"]  
print(l.index("kiwi"))
```

- **isalnum()**
  - Returns true if all characters are alphanumeric

```
str = "Sophia123"  
print(str.isalnum())
```

## METHOD DESCRIPTION EXAMPLE

METHOD	DESCRIPTION	EXAMPLE
isalpha()	Checks if its alphabetic	alpha = "what" print(alpha.isalpha()) True
isdigit()	Checks if is a digit	Digit = "1994" print(digit.isdigit()) True
islower()	Checks if its lowercase	str = "whatever" print(str.islower()) True
isnumeric()	Checks if is numeric	Str = "1234" print(str.isnumeric()) True
isspace()	Checks is its space	Space = " " print(space.isspace()) True
istitle()	Check if its a title	s = 'Python Is Good.' print(s.istitle()) True
isupper()	Check if its uppercase	s = "WHAT" print(s.isupper()) True
join()	Joins multiple things in a list	L = ["what", "is", "happening"] print(' '.join(L)) What is happening
lower()	Turns into lowercase	s = SHAPE print(s.lower())



<b>lstrip()</b>	<b>Returns the string without leading whitespace</b>	
<b>replace()</b>	<b>Returns a new string with a replacement</b>	
<b>rstrip()</b>	<b>returns a new string with trailing whitespace removed</b>	
<b>split()</b>	<b>Splits string into a list</b>	
<b>splitlines()</b>	<b>Splits a string into new lines</b>	
<b>startswith()</b>	<b>Checks if str starts with the value</b>	
<b>strip()</b>	<b>Removes any leading and trailing spaces</b>	
<b>swapcase()</b>	<b>Changes upper for lower case and viceversa</b>	

