THEORY QUESTIONS ASSIGNMENT

Python based theory

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

1. Python theory questions

30 points

-What is Python and what are its main features?
Python is an object oriented programming language python can be used to perform multiple tasks. Python can be used to conduct data analysis and it can also be used to create software programs.
-Discuss the difference between Python2 and Python3
Python 3 is preferable for software developers because it's a lot more readable. Python 3 is faster than python 2 and Python 2 uses older syntax.

-What is PEP 8?

Pep8 is a best practice document that provides guidelines on how to write python code. The document was written by Guido van Rossum, Barry Warsaw, and Nick Coghlan.

In computing/computer science what is a program?
A program is a set of instructions given to a computer to perform a certain action.

In computing / computer science what is a process?
A process is an instance of a program running in a computer. It is close in meaning to task , a term used in some operating systems.
In computing / computer science what is cache?
A cache memory, also called cache, supplementary memory system that temporarily stores frequently used instructions and data for quicker processing by the central processing unit (CPU) of a computer. Cache is a memory system that stores the most frequently used data.This is so the retrieval of data could be a lot quicker.

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

Multithreading is the program's ability to enable more than one user at a time, without needing multiple copies running on the computer

In computing / computer science what is concurrency and parallelism and what

are the differences
Concurrency means that two tasks can start,run and complete at the same time. Parallelism is when tasks run at the same time for example on a single core machine.
What is GIL in Python and how does it work?
Gil is a lock on the python interpreter that ensures that only thread can be in a state of execution.
What do these software development principles mean: DRY, KISS, BDUF
kiss is a phrase that translates to keep it simple stupid

DRY:Stands for dont repeat yourself
BDUF: Big Design Up Front (BDUF)

What is a Garbage Collector in Python and how does it work?
By deleting unwanted objects(class instances or built-in types) automatically, python frees up memory space. The process by which python periodically frees and reclaims memory blocks that are no longer in use is called garbage collection

How is memory managed in Python?
Memory management in python involves a private heap containing all python objects and data structures. The management of this private heap is ensured internally by the python memory Management.
What is a Python module?
A module in python are files that contain python code. A module can be imported to another python program.

What is docstring in Python?
A Python docstring is a string that describes a python module, class, function or method, so that programmers can understand what it does without having to read the implementation in detail.Likewise, it is a common practice to generate online(html) documentation automatically from python docstrings.

What is pickling and unpickling in Python? Example usage.
Pickling is the process of converting a python object hierarchy into a byte stream, while unpickling is the process of converting a byte stream(from a binary or bytes-like object) back into a python object hierarchy
What are the tools that help to find bugs or perform static analysis?
There are two software applications that can help you find bugs pychecker and pylint. Pychecker can perform static analysis and it can also give warnings.

How are arguments passed in Python by value or by reference? Give an example.
In python if you pass immutable arguments like integers, strings, and tuples into a function, it's called call-by-value. Parameters(arguments) in python are passed by reference,means that if you change what a parameter refers to within a function, the change will also reflect back in the calling function.

18. What are Dictionary and List comprehensions in Python? Provide examples.
List comprehensions are a more compact and elegant way of creating list than for-loops, and also allow you to create lists from existing lists. List comprehensions are constructed from brackets containing expression,which is followed by for clause, that is [item-expression for item in iterator] or [x for x in iterator], and can then be followed by further for or if clauses: [item-expression for item in iterator if conditional].

In python, dictionary comprehensions work very similar to list comprehensions, but only for dictionaries. They provide a convenient way to create a dictionary from an iterable or transform one dictionary into another.
The syntax is similar to that of list comprehension, for example {key: item-expression for item in iterator}, but note the inclusion of the expression pair (key:value).

19. What is namespace in Python?
Python namespaces. A namespace is a set of symbolically defined names that contain information about the objects each name refers to. You can think of a namespace as dictionary in which the keys are the object names and the values are the object themselves.
20. What is pass in Python?
The pass statement is used as a placeholder for future code. When the pass statement is executed, nothing happens, but you avoid getting an error when the empty code isn't allowed. Empty code is not allowed in loops, function definitions,class definitions, or if statements
21. What is unit test in Python?
An example of unit test is a test that checks a single function or component of code to make sure it performs as expected. Unit tests are an important part of regression testing to ensure that the code runs as expected when the code runs.

In python,slicing is a way to access parts of sequences such as strings,tuples and lists.You can also use slicing to modify or delete the items in mutable sequences such as lists.

23. What is a negative index in Python?
The Python programming language supports negative indexing of arrays, something that is not possible in most other programming languages. This means that index value of -1 gives the last element, and that he index of -2 gives the second last element of an array.
24. How can the ternary operators be used in python? Give an example.
The ternary operator, also known as the conditional expression, evaluates something based on whether a given condition is true or false. It was added to Python in version 2.5. It allows testing the condition in a single line, rather than a multi-line if else statement

25. What does this mean: *args, **kwargs? And why would we use it?
**kwargs and *args allow a function to take a variable length argument.*args passes a variable number of non keyworded arguments and on which operation of the tuple can be performed.

26. How are range and xrange different from one another?

Almost the same in terms of functionality, xrange and range provide a way to generate a list of integers that you can use however you like. The only difference is that the range returns a python list object, wheres xrange returns an xrange object.

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

It provides useful tools and features that make creating web applications easier in python. It is more accessible to new developers ince it allows you quick development of web applications.

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

A clustered index is a type of index that physically reorders table records to match the index, where's a non clustered index is type of index in which logical order of index matches physical stored order of the rows on the disk.

29. What is a ‘deadlock’ a relational database?

A deadlock occurs when two transactions are waiting for one another to release locks.For example, Transaction A might hold a lock on some rows in the accounts table and it needs to update some rows of the orders table to finish.

30. What is a ‘livelock’ a relational database?

As with a deadlock, the state of processes in a livelock constantly changes in relation to one another, none of which advances. livelocks are a special cases of resource starvation; the general definition only states that a specific process is nexus.

Method	Description	example
capitalize()	This method changes the first charter in the string into a capital letter	Text = “this is a text” Letter = text.capitalize() print(letter)
casefold()	This method changes any capital letter into lower case	Text = “This is a Text” String = text.casefold() print(letter)
center()	This method center aligns the string. If you include	Text = “apple” String = text.center(100) print(string)

count()	This method returns the amount of times a value appears in the string	Text = “The apple does not fall far from the apple tree” string = text.count(“apple”) print(string)
endswith()	The method returns true if the string ends with a specific value and if it doesn't it returns false	Text = “This is a text” String = text.endswith(“t”) print(String)
find()	This method finds the first instance of a value	Text = “This is a text” String = text.endswith(“t”) print(String)
format()	This method adds a value into the string	txt = "it's only {price} pounds!" print(txt.format(price = 10))
index()	The index method finds the first instance of a specific value	Text = “This is a text” String = text.index(“t”) print(String)
isalnum()	This method checks if all characters are alphanumeric	Text = "text" String = Text.isalnum() print(String)
Isalpha()	This method checks if all a characters are alphabet letters	Text = "text" String = Text.isalpha() print(String)

isdigit()	This method checks if the characters are digits.	Num = “40” String = Num.isdigit() print(String)

islower()	This method checks if all characters in a string are uppercase	Text = "text" String = Text.islower() print(String)

Isnumeric()	This method checks if all characters are numbers	Num = “40” String = Num.isnumeric() print(String)

isspace()	This method checks if all characters in a string is whitespaces	Text = “ “ String = Text.isspace() print(String)

istitle()	This method checks if each starts with a capital	Text = "This Is A Text" String = Text.istitle() print(String)

isupper()	This method checks if all characters are uppercase	Text = "THIS IS A TEXT" String = Text.isupper() print(String)

join()	This method takes all items in iterable into a one string	text = ("mia", "may", "lisa") x = " ".join(text) print(x)

lower()	This method takes a input and returns all the characters as lower case	text = "THIS IS A TEXT" string = text.lower() print(string)

istrip	This method removes any space in the beginning and the end	`text = "   text  "`
		
		`string = text.strip()`
		
		`print("this is", "a", string)`

replace()	This method replaces one one character with another	Text = “this is a text” string = Text.replace(“text”,”phrase”) print(string)
rsplit()	This method splits a string into a list.	text = "THIS IS A TEXT" string = text.rsplit() print(string)

rstrip()	This method removes any space in the beginning and the end	`text = "   text  "`
		
		`string = text.strip()`
		
		`print("this is", "a", string)`

split()	This method splits a string into a list.	text = "THIS IS A TEXT" string = text.rsplit() print(string)
splitlines()	This method splits a string into a list and separates the items with line breaks	text = "THIS IS\n A TEXT" string = text.splitlines() print(string)

startswith()	This method checks if a string starts with a specific value	text = "THIS IS A TEXT" string = text.startswith(“THIS”) print(string)

strip()	This method removes any space in the beginning and the end	`text = "  text "`
		
		`string = text.strip()`
		
		`print("this is", "a", string)`

swapcase()	This method takes a string lower case string and makes it and upper case and takes a upper case and makes it lower case	text = "tHIS iS A tEXT" string = text.swapcase() print(string)

title()	This method returns a first a string whose first character is a capital letter	Text = "this is a text" String = Text.title() print(String)

upper()	This method turns all characters in the string into uppercase	Text = "this is a text" String = Text.upper() print(String)

Python list methods:

describe each method and provide an example

Method	Description	Example
append()	This method adds a item to the end of the list	veg = ["carrots", "kale", "broccoli "] veg.append("spinach") print(veg)
clear()	This method removes all items in a list	veg = ["carrots", "kale", "broccoli "] veg.clear() print(veg)
copy()	This method returns a copy of the list	veg = ["carrots", "kale", "broccoli "] C = veg.copy() print(c)
count()	This method returns the number of times a value shows up in list	veg = ["carrots", "kale", "broccoli "] X = veg.count(“kale”) print(veg)
extend()	This method adds new values to the end of the list	veg = ["carrots", "kale", "broccoli "] fruit = [“orange”,”apple”,”mango”] veg.extend(fruit) print(veg)
index()	This method returns the position of the value	veg = ["carrots", "kale", "broccoli "] X = veg.index(“kale”)

		print(veg)

insert()	This method adds a new element onto the list. You can also specify which part you would like to add the item	veg = ["carrots", "kale", "broccoli "] veg.insert(1,“spinach”) print(veg)

pop()	This method removes an element of the list. You can also specify which part you would like to remove the item	veg = ["carrots", "kale", "broccoli "] veg.insert(1) print(veg)

remove()	This method removes items from the list.	veg = ["carrots", "kale", "broccoli "] veg.remove(“carrots”) print(veg)

reverse()	This method reverses the elements in the list	veg = ["carrots", "kale", "broccoli "] veg.reverse() print(veg)

sort()	This method sorts item in a list in ascending order	veg = ["carrots", "kale", "broccoli "] veg.sort() print(veg)

Python tuple Methods:

describe each method and provide an example

Method	Description	Example
count()	This method returns the amount of times a specific item shows up in a list	Num = (1, 1, 2,3,5) X = num.count(1) print(x)
index()	This method returns the first instance of a particular item	Num = (1, 1, 2,3,5) X = num.index(2) print(x)

Python Dictionary Methods:

describe each method and provide an example

Method	Description	Example

clear()	This method removes all elements in a dictionary	`sweet = {`
		`  "brand": "twix",`
		`  "price": 1.00`
		`}`
		
		`sweet.clear()`
		
		`print(sweet)`

copy()	This method returns a copy of the dictionary	`sweet = {`
		`  "brand": "twix",`
		`  "price": 1.00`
		`}`

		sweet.copy() print(sweet)

fromkeys()	This method creates a new dictionary with specified methods	thekeys = ('key1', 'key2', 'key3') dict = dict.fromkeys(x) print(dict)

get()	This method returns the items key	sweet = {
		"brand": "twix",
		"price": 1.00
		}
		
		X=sweet.get("brand")
		
		print(x)

items()	This method returns a view like object that contains the key-value pairs of the dictionary,listed in tuples	sweet = {
		"brand": "twix",
		"price": 1.00
		}
		
		X = sweet.items()
		
		print(x)

keys()	This method returns a view like object that contains the key-value pairs of the dictionary,listed as list	sweet = {
		"brand": "twix",
		"price": 1.00
		}
		
		X = sweet.keys()
		
		print(x)

pop()	This method removes a specified item in a dictionary	sweet = {
		"brand": "twix",
		"price": 1.00
		}
		sweet.pop("brand")

		`print(x)`

popitem()	This method removes the last item in the dictionary	`sweet = {`
		`  "brand": "twix",`
		`  "price": 1.00`
		`}`
		
		`sweet.popitem()`
		
		`print(sweet)`

Setdefault()	This method returns the value of the item with the specified key. In the instance where the key does not exist it inserts a new value to the key.	`sweet = {`
		`  "brand": "twix",`
		`  "price": 1.00`
		`}`
		
		`X = sweet.setdefault("brand","cadbury)`
		
		`print(sweet)`

update()	This method updates the dictionary to include new values	`sweet = {`
		`  "brand": "twix",`
		`  "price": 1.00`
		`}`
		`sweet.update({"choc": "milky way"})`
		
		`print(sweet)`

values()	This method returns a view like object that contains the key-value pairs of the dictionary,listed as list	`sweet = {`
		`  "brand": "twix",`
		`  "price": 1.00`
		`}`
		
		`X = sweet.values()`
		
		`print(x)`

Python set methods:

describe each method and provide an example

Method	Description	Example
add()	This method adds a item into a set	Chocolate = {"white", "milk", "dark"} Chocolate.add("toffee") print(Chocolate)
clear()	This method clears all items in a set	Chocolate = {"white", "milk", "dark"} Chocolate.clear() print(Chocolate)
copy()	This method copies the set	Chocolate = {"white", "milk", "dark"} X = Chocolate.copy() print(x)
difference()	The method contains items that only exist in oe set and not the other	x = {"apple", "banana", "cherry"} y = {"random", "food", "apple"} z = x.difference(y) print(z)
intersection()	This method returns items that exist in both sets	x = {"apple", "banana", "cherry"} y = {"random", "food", "apple"}

		z = x.intersection(y) print(z)

issubset()	This method returns true if all items in one set are present in the other set	x = {"apple", "banana", "cherry"} y = {"random", "food", "apple"} z = x.issubset(y) print(z)

issuperset()	This method returns true if all the items in the second set are present in first set	x = {"apple", "banana", "cherry"} y = {"random", "food", "apple"} z = x.issuperset(y) print(z)

pop()	This method removes a random item in the set	Chocolate = {"white", "milk", "dark"} Chocolate.pop() print(chocolate)

remove()	This method removes a specific item in the set	Chocolate = {"white", "milk", "dark"} Chocolate.remove(“white”) print(chocolate)

symmetric_difference()	This method returns all the items in the set except for the items present in both sets	x = {"apple", "banana", "cherry"} y = {"random", "food", "apple"} z = x.intersection(y) print(z)

union()	This method returns a set	x = {"apple", "banana",

	containing a set containing all items	"cherry"} y = {"random", "food", "apple"} z = x.union(y) print(z)

update()	This method updates one set by including items from another	x = {"apple", "banana", "cherry"} y = {"random", "food", "apple"} z = x.update(y) print(z)

Python file methods:

describe each method and provide an example

Method	Description	Example

read()	This method in returns a specific number of bytes from a file	f = open("testfile.txt", "r") print(f.read())

readline()	This method returns the first line in the file	f = open("testfile.txt", "r") print(f.readline())

readlines()	This method returns a list that contains each line in the file	f = open("testfile.txt", "r") print(f.readlines())

write()	This method writes to an existing file.	`f = open("testfile.txt", "a")`
		`f.write("test!")`
		`f.close()`

writelines()	This method writes item of a list into a file	`f = open("testfile.txt", "a")`
		`f.writelines(["hello ", "bye."])`
		`f.close()`