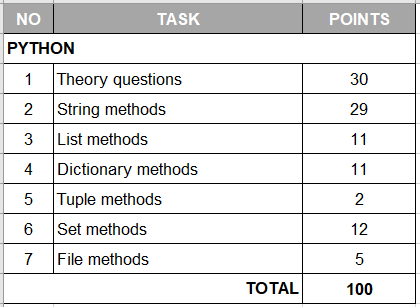
**THEORY QUESTIONS ASSIGNMENT**

Python based theory

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



|  |  |
| --- | --- |
| 1. **Python theory questions** | **30 points** |

1. What is Python and what are its main features?

A computer programming language with a feature to improve code readability, use simple syntax and have access to an availability of modules and libraries which extend the programme capabilities. It uses an object-oriented approach.

1. Discuss the difference between Python 2 and Python 3

Python 2 and python 3 are incompatible. Python 3 is the newer version of the python computer programming language. Python 3 was designed to be easier to understand and as such the syntax is simpler. In addition, Python 3 supports newer programming techniques such as data science, machine learning and artificial intelligence. Furthermore, Python 3 is mixable with other languages.

1. What is PEP 8?

A python style guide document. It helps improve the consistency and readability of python code.

1. In computing / computer science what is a program?

A specific set of ordered operations for a computer to perform to reach a solution

1. In computing / computer science what is a process?

A program in execution by one or many threads

1. In computing / computer science what is cache?

Temporary memory which stores a subset of data. It is usually used to help websites, browsers, and apps load faster.

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

A thread is a part of a program. Multithreading allows two or more threads to be executed.

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

Concurrency is the arrangement of independently executing processes (i.e. threads) for a specified time. The execution is coordinated and managed to independent lines of code and thus the executions can be parallel. It is used to make programs more usable.

Parallelism is the arrangement of many executing processes (i.e. threads) for a specified time. The execution is carried out cooperatively. It is used to make programs faster.

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

GIL – global interpreter lock. It allows one only thread to execute at a time by locking the entire interpreter when in use. This means that it is not possible for another thread to conflict with the current thread.

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

DRY – don’t repeat yourself. This means every logic/function must have a single and clear representation.

KISS – keep it simple, stupid. This means keep the system as simple as possible.

BDUF – big design up front. This means the project design should be completed first

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

The garbage collector is an automated process (algorithm) to deallocate objects that are no longer needed. It does this by reference counting and generational garbage collection

1. How is memory managed in Python?

Python uses some memory for internal use and non-object memory. The remaining memory is dedicated to object storage

1. What is a Python module?

A module is a file containing executable statements and function definitions

1. What is docstring in Python?

A string that is written as the first statement in a method definition, class, function or module

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

Pickling – converting python object into a byte stream

Unpickling – retrieving original python objects from a byte stream to object

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

Pychecker and Pylint

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

Arguments such as numbers, strings or tuples passed to a function cannot be changed outside the function as an immutable objected is passed to the function. E.g.

def add\_name(welcome\_message):

welcome\_message = welcome\_message+"{}".format('Lola')

print("Inside Function", welcome\_message)

# messages

welcome\_message = 'Warm Welcome '

add\_name(welcome\_message)

print("Outside Function:", welcome\_message)

Answers - Inside Function Warm Welcome Lola

Outside Function: Warm Welcome

Whereas, immutable objects passed to a function will have their values changed out of a function. E.g.

def add\_name(test\_result\_list):

test\_result\_list.append(100)

print("Inside Function", test\_result\_list)

# messages

test\_result\_list = [50,80,70,60]

add\_name(test\_result\_list)

print("Outside Function:", test\_result\_list)

Answers- Inside Function [50, 80, 70, 60, 100]

Outside Function: [50, 80, 70, 60, 100]

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

A dictionary is a collection of data values stored as key:value pairs. A dictionary is ordered, changeable and doesn’t allow duplicates e.g. book = {“publisher” : “collins”, ”price”:86.3}

Like dictionaries, a list is ordered, changeable and doesn’t allow duplicates. A list is used to store multiple items in a single variable e.g. book = [“collins”, “90”]

1. What is namespace in Python?

An identifier given to objects when created in python programming.

1. What is pass in Python?

A pass is a null statement, when used a ‘no operation’ is performed. It used as a placeholder in programming, typically for something that will be implemented in the future.

1. What is unit test in Python?

A test that checks a simple component is operating as required. Many unit tests are used to test an application and if designed correctly can quickly isolate what is broken in an application.

1. In Python what is slicing?

As a string is an array of characters. An index (i.e. the value in an array) is the value location within an array. Slicing function looks at the given starting index and ending index and returns a range of characters within the index.

1. What is a negative index in Python?

A negative index is when the value location within an array starts from where the array ends

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

These are operators that test if a condition is true or false and they can be used to replace a long if/else statement

# operator to see if customer gets discount based on age  
  
ticket\_price = 10.00  
discount\_if\_under\_16 = 0.85 \* ticket\_price  
  
customer\_age = int(input("Enter customer age for discount check..."))  
  
age\_condition = 16  
  
operator = discount\_if\_under\_16 if customer\_age < age\_condition else ticket\_price  
  
print(operator)

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

\*\*kwargs allows keyworded variable of arguments to a function. It is used to handle named arguments in a function

\*args allows non-keyworded variable arguments to a function. It is used to allow you to pass unspecified number of arguments to a function

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

In range the entire list is allocated to memory whereas in xrange doesn’t generate the whole range sequence as it calculates as needed requiring less memory

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

Flask is a micro web framework used for developing web applications

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

Clustered index defines the order in which data is stored in a table. The index is stored in the leaf nodes.

A non-clustered index does not sort the data inside the table. The index is located in a separate location to the table data.

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

Two concurrent transactions cannot make progress and are waiting for one to release a lock

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

A request for an exclusive lock is repeatedly denied because shared locks are interfering with each other.

|  |  |
| --- | --- |
| 1. **Python string methods:**   **describe each method and provide an example** | **29 points** |

|  |  |  |
| --- | --- | --- |
| **METHOD** | **DESCRIPTION** | **EXAMPLE** |
| capitalize() | Returns the first index/character of a string to a capital letter | s = 'lola'.capitalize()  print(s)  Answer: Lola |
| casefold() | Returns an entire string in lowercase and removes any case distinctions present so strings can be caseless matched | s = 'Rußen'.casefold()  print(s)  Answer: russen |
| center() | Returns a string with centre alignment against a specified character length | s = "Lola's work"  text\_centre = s.center(20)  print(s)  Answer: ' Lola\'s work ' |
| count() | Provides the number of elements within the value searched | s = "Lola's work"  text = s.count('o')  print(text)  Answer: 2 |
| endswith() | A method which returns true or false based on the specified value of a string end | s = "Lola's work"  text = s.endswith('work')  print(text)  Answer: true |
| find() | A method which returns either the first occurrence of a value or if not found returns -1. This varies from index() method which raises an exception if the value is not found | s = "Lola's work"  text = s.find('a')  print(text)  Answer: 3 |
| format() | A method which returns a formatted string. The string is formatted by inserting values inside the string’s placeholder | text = 'work'  s = "Lola's {}".format(text)  print(s)  Answer – Lola’s work |
| index() | A method which returns either the first occurrence of a value or if not found raises an exception | s = "Lola's work"  text = s.index('w')  print(text)  Answer: 7 |
| isalnum() | A method that checks (and returns true or false) if all characters in a string are alphanumbers i.e. only containing alphabet letters and numbers | s = "Lola's work &&&&"  text = s.isalnum()  #false expected as symbol added  print(text)  Answer: false |
| isalpha() | A method that checks (and returns true or false) if all characters in a string are alphabet letters (a-z) | s = "Lola's work 123"  text = s.isalpha()  #false expected as numbers added  print(text)  Answer: false |
| isdigit() | A method that checks if all characters in a string are digits. Digits include numeric characters, superscript and subscript. Digit characters exclude fractions and roman numerals. | s = "123"  text = s.isdigit()  #true expected  print(text)  Answer: true |
| islower() | A method which returns true if all alphabet characters within a string are lower case | s = "Lola's work"  text = s.islower()  #false expected as L is captial  print(text)  Answer: False |
| isnumeric() | A method which returns true if all characters within a string are numeric. Numeric includes decimal, digit and numeric (including fraction, roman numeral and currency numerators | s = "9029"  text = s.isnumeric()  #True expected only numeric used  print(text)  Answer: True |
| isspace() | A method which returns true is all characters within a string are whitespaces. | s = " "  text = s.isspace()  #True expected as no letters/numbers  print(text)  Answer: True |
| istitle() | A method which returns true if each word in the string start with an uppercase letter and the remaining letters in the word are lowercase | s = 'Lola Work'  print(s.istitle())  Answer: true |
| isupper() | A method which returns true if letter in each word is an uppercase letter | s = 'LOLA WORK'  print(s.isupper())  Answer: true |
| join() | A method which takes all items and joins them into one string with a defined separator | s = ("lola", "Peter", "Mary")  x = ",".join(s)  print(x)  Answer: lola,Peter,Mary |
| lower() | A method which converts all characters in a string to lowercase. | s = 'LOLA WORK'  print(s.lower())  Answer: lola work |
| lstrip() | A method which removes the specified characters from the start of the string until a different character to that specified is reached | s = 'lola work'  x = s.strip('l')  print(x)  Answer: ola work |
| replace() | A method to replace a specified phrase with another specified phrase | s = 'lola work'  x = s.replace("work", "work work")  print(x)  Answer - lola work work |
| rsplit() | A method to split a string into a list starting from right and using a defined separator | s = 'lola $ work'  x = s.rsplit("$")  print(x)  Answer - ['lola ', ' work'] |
| rstrip() | A method that removes whitespaces at end of a string. | s = "lola work "  x = s.rstrip()  print(x) |
| split() | A method to split a string into a list where each word is a list item using a defined separator. By default, separator is any whitespace | s = "lola work for theory "  x = s.split()  print(x)  Answer - ['lola', 'work', 'for', 'theory'] |
| splitlines() | A method to split a string into a list where each line is a list item. | s = "lola work for theory \n theory assignment "  x = s.splitlines()  print(x)  Answer: ['lola work for theory ', ' theory assignment '] |
| startswith() | A method which returns true or false if a string starts with a specified value | s = "lola work for theory assignment "  print(s.startswith('work'))  Answer: False |
| strip() | A method that removes whitespaces at the beginning and end of a string. | s = " Lola Work "  x = s.strip()  print(x) |
| swapcase() | A method that swaps the uppercase letters with the lowercase letters and the lowercase letters with uppercase letters. | s = 'LOlA WoRk'  print(s.swapcase())  Answer: loLa wOrK |
| title() | A method that returns a string with all characters in title formatting i.e. the first character in every word is upper case | s = "my homework" x = s.title() print(x)  Answer: My Homework |
| upper() | A method that returns a string with all characters in upper case | s = "my homework"  x = s.upper()  print(x)  Answer: MY HOMEWORK |

|  |  |
| --- | --- |
| 1. **Python list methods:**   **describe each method and provide an example** | **11 points** |

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| [append()](https://www.w3schools.com/python/ref_list_append.asp) | A method which adds a single element to the end of the list. List increases by one | s = ['Lola']  s.append("Work")  print(s)  Answer: [‘Lola', 'Work'] |
| [clear()](https://www.w3schools.com/python/ref_list_clear.asp) | A method which removes all items from a list | s = ['Lola','Work']  s.clear()  print(s)  Answer: [] |
| [copy()](https://www.w3schools.com/python/ref_list_copy.asp) | A method which creates a copy of an existing list and returns the copy of the copied list in a new list | s = ['Lola','work']  n = s.copy()  print(n)  Answer: ['Lola', 'work'] |
| [count()](https://www.w3schools.com/python/ref_list_count.asp) | Provides the number of elements within the list searched against a specified element | s = ['Lola','work','is','by','Lola']  count = s.count('Lola')  print(count)  Answer: 2 |
| [extend()](https://www.w3schools.com/python/ref_list_extend.asp) | A method which adds elements to the end of the list. List extends by number of elements in an iterable argument | s = ['Lola','work']  n = ['is','by','Lola']  s.extend(n)  print(s)  Answer: ['Lola', 'work', 'is', 'by', 'Lola'] |
| [index()](https://www.w3schools.com/python/ref_list_index.asp) | A method which returns either the first occurrence of a value in a list or if not found raises an exception | s = ['Lola','work']  index = s.index('work')  print(index)  Answer : 1 |
| [insert()](https://www.w3schools.com/python/ref_list_insert.asp) | A method which inserts a value at a specified position within a list | s = ['Lola','work']  insert = s.insert(1, 'great')  print(s)  Answer : ['Lola', 'great', 'work'] |
| [pop()](https://www.w3schools.com/python/ref_list_pop.asp) | A method which removes a value at a specified position within a list and returns the removed value | s = ['Lola','work']  pop = s.pop(0)  print(pop)  Answer: [‘Lola’] |
| [remove()](https://www.w3schools.com/python/ref_list_remove.asp) | A method which removes the first matching value within a list and returns the list without the removed value | s = ['Lola','work']  s.remove('work')  print(s)  Answer: [‘Lola’] |
| [reverse()](https://www.w3schools.com/python/ref_list_reverse.asp) | A method which reverses the order of the elements in a list | s = ['Lola','work']  s.reverse()  print(s)  Answer: ['work', 'Lola'] |
| [sort()](https://www.w3schools.com/python/ref_list_sort.asp) | A method which sorts the list in ascending order by default, however there is a key option to specify the sorting criteria | s = ['Lola','work', 'A']  s.sort()  print(s)  Answer: ['A', 'Lola', 'work'] |

|  |  |
| --- | --- |
| 1. **Python tuple methods:**   **describe each method and provide an example** | **2 points** |

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| [count()](https://www.w3schools.com/python/ref_tuple_count.asp) | Provides the number of elements within the tuple searched | s = ('l','o','l','a')  count = s.count('l')  print(count)  Answer: 2 |
| [index()](https://www.w3schools.com/python/ref_tuple_index.asp) | A method which returns either the first occurrence of a value in a tuple or if not found raises an exception | s = ('l','o','l','a')  index = s.index('a')  print(index)  Answer:3 |

|  |  |
| --- | --- |
| 1. **Python dictionary methods:**   **describe each method and provide an example** | **11 points** |

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| [clear()](https://www.w3schools.com/python/ref_dictionary_clear.asp) | A method which removes all items from a dictionary | s = {  "name": "lola",  "task": "work"  }  s.clear()  print(s)  Answer: {} |
| [copy()](https://www.w3schools.com/python/ref_dictionary_copy.asp) | A method which creates a copy of an existing dictionary and returns a new dictionary which has the copied dictionary | s = {  "name": "lola",  "task": "work"  }  n = s.copy()  print(n)  Answer: {'name': 'lola', 'task': 'work'} |
| [fromkeys()](https://www.w3schools.com/python/ref_dictionary_fromkeys.asp) | A method which creates a new dictionary from elements used as keys and values if specified (otherwise ‘None’ added for values) | k = {'name','task'}  v = {'value'}  new = dict.fromkeys(k, v)  print(new)  Answer: {'name': {'value'}, 'task': {'value'}} |
| [get()](https://www.w3schools.com/python/ref_dictionary_get.asp) | A method which returns they value of the specified key item | s = {  "name": "lola",  "task": "work",  }  x = s.get("name")  print(x)  Answer: lola |
| [items()](https://www.w3schools.com/python/ref_dictionary_items.asp) | A method which returns the list view of a dictionary as a tuple pair (key,value). | s = {  "name": "lola",  "task": "work"  }  print(s.items())  Answer: dict\_items([('name', 'lola'), ('task', 'work')]) |
| [keys()](https://www.w3schools.com/python/ref_dictionary_keys.asp) | A method which returns the key view of a dictionary as a list. Values are not included in this view and changes made to dictionary are reflected in this view | s = {  "name": "lola",  "task": "work"  }  print(s.keys())  Answer: dict\_keys(['name', 'task']) |
| [pop()](https://www.w3schools.com/python/ref_dictionary_pop.asp) | A method which removes a value at a specified position within a dictionary using the given key and returns the removed value | s = {  "name": "lola",  "task": "work"  }  name = s.pop('name')  print(name)  Answer: lola |
| [popitem()](https://www.w3schools.com/python/ref_dictionary_popitem.asp) | A method which removes the last item in a dictionary and returns the removed item | s = {  "name": "lola",  "task": "work"  }  s.popitem()  print(s)  Answer: {'task': 'work'} |
| [setdefault()](https://www.w3schools.com/python/ref_dictionary_setdefault.asp) | A method which returns the value of the specified key and if not found then inset the specified key and specified value | s = {  "name": "lola",  "task": "work"  }  d = s.setdefault("cohort", "software2")  print(s)  Answer: {'cohort': 'software2', 'name': 'lola', 'task': 'work'} |
| [update()](https://www.w3schools.com/python/ref_dictionary_update.asp) | A method which adds elements to the end of the dictionary. Dictionary extends by number of elements in iterable argument | s = {  "name": "lola",  "task": "work",  }  update = {  "cohort": "software2"  }  s.update(update)  print(s)  Answer: {'cohort': 'software2', 'name': 'lola', 'task': 'work'} |
| [values()](https://www.w3schools.com/python/ref_dictionary_values.asp) | A method which returns the value view of a dictionary as a list. Keys are not included in this view and changes made to dictionary are reflected in this view | s = {  "name": "lola",  "task": "work"  }  print(s.values())  Answer: dict\_values(['lola', 'work']) |

|  |  |
| --- | --- |
| 1. **Python set methods:**   **describe each method and provide an example** | **12 points** |

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| [add()](https://www.w3schools.com/python/ref_set_add.asp) | A method which adds an element to a specified set | s = {"Lola", "work"}  s.add("great")  print(s)  Answer: {'work', 'great', 'Lola'} |
| [clear()](https://www.w3schools.com/python/ref_set_clear.asp) | A method which removes all items from a set | s = {"Lola", "work"}  s.clear()  print(s)  Answer: set() |
| [copy()](https://www.w3schools.com/python/ref_set_copy.asp) | A method which creates a copy of a set and returns the copy in a new set | s = {"Lola", "work"}  n = s.copy()  print(n)  Answer: {'work', 'Lola'} |
| [difference()](https://www.w3schools.com/python/ref_set_difference.asp) | A method that looks at the differences between two sets and return items from the first set that are not in the second set | s = {"Lola", "work"}  d = {"Lola", "play"}  n = s.difference(d)  print(n)  Answer: {'work'} |
| [intersection()](https://www.w3schools.com/python/ref_set_intersection.asp) | A method that looks at two or more sets and returns a set that contains similarities between the sets | s = {"Lola", "work"}  d = {"Lola", "play"}  n = s.intersection(d)  print(n)  Answer: {'Lola'} |
| [issubset()](https://www.w3schools.com/python/ref_set_issubset.asp) | A method which returns true if all elements in existing set is in a specified set. If not, false is returned | s = {"Lola", "work", "great"}  d = {"Lola", "work", "great"}  x = s.issubset(d)  print(x)  Answer: True |
| [issuperset()](https://www.w3schools.com/python/ref_set_issuperset.asp) | A method which returns true if all elements in a set is in an original set. If not, false is returned | s = {"Lola", "work", "great"}  d = {"Lola"}  x = s.issuperset(d)  print(x)  Answer: True |
| [pop()](https://www.w3schools.com/python/ref_set_pop.asp) | A method which removes a random value within a set and returns the removed value | s = {"Lola", "work", "great"}  x = s.pop()  print(x) |
| [remove()](https://www.w3schools.com/python/ref_set_remove.asp) | A method that removes a specified element from a set and returns the set without the removed element | s = {"Lola", "work"}  s.remove('Lola')  print(s)  Answer: {'work'} |
| [symmetric\_difference()](https://www.w3schools.com/python/ref_set_symmetric_difference.asp) | A method that looks at the differences between two sets and returns the differences from both sets | s = {"Lola", "work"}  d = {"Lola", "play"}  n = s.symmetric\_difference(d)  print(n)  Answer: {'work', 'play'} |
| [union()](https://www.w3schools.com/python/ref_set_union.asp) | A method that combines all elements from specified sets and removes duplicates which appear. | s = {"Lola", "work", "great"}  d = {"cool","Lola", "work"}  x = s.union(d)  print(x)  Answer: {'work', 'Lola', 'cool', 'great'} |
| [update()](https://www.w3schools.com/python/ref_set_update.asp) | A method that updates an original set by adding iterable items from another set. If iterable items are present in both sets then only one appearance of the item will be present in the updated set. | s = {"Lola", "work", "great"}  d = {"cool","nice", "work"}  s.update(d)  print(s)  Answer: {'work', 'nice', 'Lola', 'cool', 'great'} |

e

|  |  |
| --- | --- |
| 1. **Python file methods:**   **describe each method and provide an example** | **5 points** |

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| [read()](https://www.w3schools.com/python/ref_file_read.asp) | A method which returns a specified number of bytes from the specified file. By default, the whole file is specified in the number of bytes | f = open("example.txt", "r")  print(f.read()) |
| [readline()](https://www.w3schools.com/python/ref_file_readline.asp) | A method which returns one line in the specified file | f = open("example.txt", "r")  print(f.readlines()) |
| [readlines()](https://www.w3schools.com/python/ref_file_readlines.asp) | A method which returns a list containing each line in the specified file as a list item | f = open("example.txt", "r")  print(f.readlines()) |
| [write()](https://www.w3schools.com/python/ref_file_write.asp) | A method which writes text to a specified file | f = open("example.txt", "r")  f.write(“Lola work”) |
| [writelines()](https://www.w3schools.com/python/ref_file_writelines.asp) | A method which writes text to a specified file as items of a list | f = open("example.txt", "r")  f.writelines([“Lola work”, “cool work”.]) |