**THEORY QUESTIONS ASSIGNMENT**

**Baimaganbetova Madina**

**Python based theory**

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

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

Python is a programming language. The main features of Python are: object oriented language, simple syntax(easy to read/write/learn), many available libraries (especially good for data analysis), it is interpreted language, it is open source, and it can be used on different

**2. Discuss the difference between Python 2 and Python 3**

Python 3 has way more libraries than python 2. Several syntaxis changes (like print function). Strings are stored as ASCII by default in python2, and UNICODE in python3; division in python2 yields in int, and python3-float;

**3. What is PEP 8?**

It is a guidelines and recommendations on how python code should look like. Some sort of best practices

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

Program is a list of instructions written in special language so that a computer can interpret and execute it

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

A process is a self-contained task in the OS, which can run concurrently with other processes.

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

Cache is usually a small amount of memory which has much faster access speed compared to RAM, because it's closer to the CPU. Some values can be stored in cache (cached), making their access faster.

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

A thread execution is an execution of a sequence of instructions that can be "interrupted" to start executing other threads. This allows one CPU to be concurrent and creates and illusion of multitasking, whereas it's actually very fastly switches between different thread executions. And multithreading is when one process spawns multiple threads for running, for example, a complex application.

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

Concurrency is a way to switch between tasks fastly to create an illusion of them being executed at the same time. One CPU can execute only one instruction at an instance. On the other hand, parallelism actually runs multiple tasks/instructions at an instance, and usually achieved by multi-core processors.

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

Python's Global Interpreter Lock makes sure that there are no and deadlocks (e.g. garbage collection) by putting a lock on the interpreter, so that only one thread can be run at one single point of time.

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

**DRY -** Don't Repeat Yourself. Principle aimed to eliminate repetition in code where possible

**KISS –** Keep it simple stupid. Principle aimed to simplify the code where possible

**BDUF –** Big design up front. Principle aimed development and improvement of  design before implementation stage of the project

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

Python remove unnecessary objects to free the memory space automatically. This process of memory cleaning reclaims blocks of memory that are not used is called Garbage Collection.

**12. How is memory managed in Python?**

Python has a special object allocator that allocates memory. The object allocator gets a call every time the new object needs new space. Additionally, Python uses a part of the memory for objects of different types and non-object memory and internal use.

**13. What is a Python module?**

Python modules are .py files that consist of Python code. Any Python file can be referenced as a module

**14. What is docstring in Python?**

Python docstrings are the strings that usually contain some information and explanations of the functions and located right after the definition of a function, method, class, or module.

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

“Pickling” is the process when Python object is converted into a byte stream, and “unpickling” is the inverse operation.

import pickle

#pickling

wizards = ['Harry','Ron','Draco','Albus']

with open('wizards.pkl','wb') as wizardspkl:

    pickle.dump(wizards,wizardspkl)

#unpickling

with open('wizards.pkl','rb') as wizardspkl:

    withards = pickle.load(wizardspkl)

print(wizards)

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

Some examples of tools that could be used for static analysis Mypy, Pylint, Pyflakes, etc. These tools help to detect errors in code and implement style checks according to PEP-8 standard

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

In python arguments can be called both by value and reference. If you pass immutable arguments to a function (int, strings, tuple), the passing acts like pass by value.

If we pass mutable arguments, they are passed by reference.

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

Both Dictionary and List comprehensions are data structures, which are very similar to ordinary lists and dictionaries, but contain loops within their syntax. Therefore require much less code.

List comprehension:

conseq\_nums = [num for num in range(10)]

print(conseq\_nums)

output: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Dictionary comprehension:

dict\_1 = {0:0, 1:1, 2:2, 3:3}

dict = {key + 1: value - 1 for (key,value) in dict\_1.items()}

print(dict)

output: {1: -1, 2: 0, 3: 1, 4: 2}

**19. What is namespace in Python?**

A namespace is a collection of defined symbolic names coming with information about the object that each name references.

**20.What is pass in Python?**

Pass is a statement that does nothing, it is some kind of NULL operation, that does not return any result

**21. What is unit test in Python?**

It is a type of testing when the small parts of the program are tested separately, to ensure their correct performance

**22. In Python what is slicing?**

Slicing is a way to access parts of the strings, lists or other sequences. In order to slice one needs to pass starting and ending position of the required part

**23. What is a negative index in Python?**

Negative index is indexing in opposite direction, from end to start. [-1] refers to the last item in the list/string/other iterable object

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

Ternary operators are conditional expressions that checks whether the condition is true or not within a single line.

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

\*args refer to the arguments of any amount, \*args are referred by value and used when declaring a function

\*kwargs similarly refer to any number of arguments, but passed as keyword. The keywords and their values are saved as key-value pair in the dictionary.

Both of this arguments can be used when we are unsure about the number of arguments that is going to be passed to the function

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

The range and xrange are both functions used to iterate within for loop for a given number of times. Depending on the version of Python the functionality is slightly different.

Python 2:

range(n) – returns a list with n elements

xrange(n) – returns an object that can be used to display the elements within loop

Python 3:

xrange(n) – depreciated

range(n) – behaves very similar to xrange in Python 2

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

Flask is a lightweight web framework used to build APIs/web applications using Python. Flask can be used to develop web applications of different complexity (for example, binding to databases and frontend)

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

Clustered index is a type of index when records are matching the indexes in specific order, it can be initiated when the table is created with the primary key. The rows order of clustered indexes in the tables is similar to the order of rows in the data table

Non-clustered indexes consist of list of references that point to the data. The order of the rows of indexes might not coincide with the original data table

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

A deadlock is a situation when several transactions are not completed because they wait for each another to give up locks. If the two transactions hold locks on the data required for the other transaction, the transactions might last forever, until one of the is terminated, and second one can proceed

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

A livelock in contrast to deadlock is a situation when multiple different threads can't acquire lock because they keep interfering with each other when one of them tries to lock it

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

|  |  |  |
| --- | --- | --- |
| **METHOD** | **DESCRIPTION** | **EXAMPLE** |
| **capitalize()** | **Returns a string with first letter capital and the remaining letters are lowercased** |  |
| **casefold()** | **Returns a string with all characters lowercased.** |  |
| **center()** | **Returns a string of given length aligned in the center. The length of the returned output needs to be provided as a parameter. Additionally, the space on each side of the string can be filled with specified character.** |  |
| **count()** | **Returns number of times the specific substring (or character) appears within a string.** |  |
| **endswith()** | **Checks whether the string ends with the specified substring (or character), returns True/False.** |  |
| **find()** | **Returns the index of first occurrence of the specified substring. Additionally, we can also specify the starting and ending indexes. If the substring is not found the method returns -1** |  |
| **format()** | **Returns a formatted string with the specified variable value inserted inside the brackets.** |  |
| **index()** | **Searches the string for a specified value and then returns its index if it was found. Optionally, start and end positions can be specified.** |  |
| **isalnum()** | **Checks whether the string consists of only alphanumeric characters, and returns a Boolean (True/False)** |  |
| **isalpha()** | **Checks whether the string consists of only letter characters, and returns a Boolean (True/False)** |  |
| **isdigit()** | **Checks whether the string consists of only numeric characters, and returns a Boolean (True/False)** |  |
| **islower()** | **Checks whether the string consists of only lowercase characters, and returns a Boolean (True/False)** |  |
| **isnumeric()** | **Checks whether the string consists of only numeric characters, and returns a Boolean (True/False)** |  |
| **isspace()** | **Checks whether the string consists of only whitespace characters, and returns a Boolean (True/False)** |  |
| **istitle()** | **Checks whether each word of the string starts with capital letter and the rest of the word is lowercase letters. The method returns a Boolean value.** |  |
| **isupper()** | **Checks whether the string consists of all uppercase characters and returns a Boolean value.** |  |
| **join()** | **Joins all elements of iterable object, such as list or tuple, into a string using provided separator.** |  |
| **lower()** | **Converts all letters in a string to lower case.** |  |
| **lstrip()** | **Removes blank spaces in the beginning of the string. Optionally can remove any specified character in the beginning of the string.** |  |
| **replace()** | **Replaces a specified string with another value. Additionally, the number of replacing occurrences might be specified.** |  |
| **rsplit()** | **Splits the string with specified separator into a list of strings starting from the right.** |  |
| **rstrip()** | **Removes all the specified characters at the end of the string. Space is the default character to remove.** |  |
| **split()** | **Splits the string into a list using a specified separator starting from the left.** |  |
| **splitlines()** | **Splits the string into a list at line breaks.** |  |
| **startswith()** | **Checks if the string starts with the specified charachter or not. Returns boolean** |  |
| **strip()** | **Removes the specified characters from the beginning and end of the given string. Whitespace is the default character to remove.** |  |
| **swapcase()** | **Changes all lower case characters to uppercase and vice versa.** |  |
| **title()** | **Changes every first character of each word to upper case** |  |
| **upper()** | **Changes all the characters in the string to upper case.** |  |

|  |  |
| --- | --- |
| 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) | **Takes a single argument and adds it to the end of the list.** |  |
| [**clear()**](https://www.w3schools.com/python/ref_list_clear.asp) | **Removes all elements from the list** |  |
| [**copy()**](https://www.w3schools.com/python/ref_list_copy.asp) | **Creates a copy of the given list** |  |
| [**count()**](https://www.w3schools.com/python/ref_list_count.asp) | **Counts the number of appearances of the specified item.** |  |
| [**extend()**](https://www.w3schools.com/python/ref_list_extend.asp) | **Appends list of items to another list** |  |
| [**index()**](https://www.w3schools.com/python/ref_list_index.asp) | **Searches the list for a specified value and then returns index of the first occurrence of this item if it was found.** |  |
| [**insert()**](https://www.w3schools.com/python/ref_list_insert.asp) | **Inserts given element to the specified position in the list** |  |
| [**pop()**](https://www.w3schools.com/python/ref_list_pop.asp) | **Removes the last element from the list. Optionally the certain position of the element can be specified. The function return the removed element** |  |
| [**remove()**](https://www.w3schools.com/python/ref_list_remove.asp) | **Removes the specified item from the list** |  |
| [**reverse()**](https://www.w3schools.com/python/ref_list_reverse.asp) | **Reverses order of the elements in the list.** |  |
| [**sort()**](https://www.w3schools.com/python/ref_list_sort.asp) | **Sorts the elements of similar type in the list in ASCII order.** |  |

|  |  |
| --- | --- |
| 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) | **Returns the number of given elements in the tuple.** |  |
| [**index()**](https://www.w3schools.com/python/ref_tuple_index.asp) | **Returns the index position of the given element in the tuple.** |  |

|  |  |
| --- | --- |
| 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) | **Deletes all the elements in the dictionary** |  |
| [**copy()**](https://www.w3schools.com/python/ref_dictionary_copy.asp) | **Creates a copy of the dictionary** |  |
| [**fromkeys()**](https://www.w3schools.com/python/ref_dictionary_fromkeys.asp) | **Creates a dictionary with provided list of keys and one value** |  |
| [**get()**](https://www.w3schools.com/python/ref_dictionary_get.asp) | **Returns the value of the specified key** |  |
| [**items()**](https://www.w3schools.com/python/ref_dictionary_items.asp) | **Returns a dict\_items object with list of tuples of key and value pairs.** |  |
| [**keys()**](https://www.w3schools.com/python/ref_dictionary_keys.asp) | **Returns a dict\_keys object with list of keys from the dictionary** |  |
| [**pop()**](https://www.w3schools.com/python/ref_dictionary_pop.asp) | **Removes the key-value pair from the dictionary by key. And returns value of the removed key-value** |  |
| [**popitem()**](https://www.w3schools.com/python/ref_dictionary_popitem.asp) | **Removes the last added itemin the dictionary and returns the removed pair** |  |
| [**setdefault()**](https://www.w3schools.com/python/ref_dictionary_setdefault.asp) | **Adds given key-value pair and returns the value if the given key is not in the dictionary. If the given key exists, simply returns the value of the given key.** |  |
| [**update()**](https://www.w3schools.com/python/ref_dictionary_update.asp) | **Updates the dictionary with the specified key-value pair.** |  |
| [**values()**](https://www.w3schools.com/python/ref_dictionary_values.asp) | **Returns dict\_values object with list of values from the dictionary.** |  |

|  |  |
| --- | --- |
| 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) | **Adds element to the set. If the element is already in the set, the duplicates are not created** |  |
| [**clear()**](https://www.w3schools.com/python/ref_set_clear.asp) | **Deletes all the elements in the set.** |  |
| [**copy()**](https://www.w3schools.com/python/ref_set_copy.asp) | **Creates the copy of the set.** |  |
| [**difference()**](https://www.w3schools.com/python/ref_set_difference.asp) | **Returns a set of elements that are in set 1 but not in the set 2.** |  |
| [**intersection()**](https://www.w3schools.com/python/ref_set_intersection.asp) | **Returns the set of common values for two given sets** |  |
| [**issubset()**](https://www.w3schools.com/python/ref_set_issubset.asp) | **Checks if one set is a subset of another set, returns Boolean** |  |
| [**issuperset()**](https://www.w3schools.com/python/ref_set_issuperset.asp) | **Checks if one set contains all the elements of the second set, and returns boolean** |  |
| [**pop()**](https://www.w3schools.com/python/ref_set_pop.asp) | **Removes one element from the set, and returns the removed element** |  |
| [**remove()**](https://www.w3schools.com/python/ref_set_remove.asp) | **Removes the given element from the set.** |  |
| [**symmetric\_difference()**](https://www.w3schools.com/python/ref_set_symmetric_difference.asp) | **Returns set of elements which are not common for two given sets** |  |
| [**union()**](https://www.w3schools.com/python/ref_set_union.asp) | **Returns a single set with all the elements from given sets.** |  |
| [**update()**](https://www.w3schools.com/python/ref_set_update.asp) | **Adds missing elements to one set from another** |  |

|  |  |
| --- | --- |
| 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) | **Reads the text from the file.** |  |
| [**readline()**](https://www.w3schools.com/python/ref_file_readline.asp) | **Read the text from file line by line.** |  |
| [**readlines()**](https://www.w3schools.com/python/ref_file_readlines.asp) | **Returns a list of lines of the text from the file.** |  |
| [**write()**](https://www.w3schools.com/python/ref_file_write.asp) | **Write text to the file. Writing file from scratch by specifying ‘w’. Adding text to the existing text in the file by specifying ‘a’ in the open method.** | with open('Assignment1.txt', 'a') as file:      file.writelines(['''\nAnd there's a cold lonely light that shines from you\n''', '''You'll wind up like the wreck you hide behind that mask you use''']) |
| [**writelines()**](https://www.w3schools.com/python/ref_file_writelines.asp) | **Write text to the file line by line from the given list of lines. Similarly has ‘w’ or ‘a’ specifications in the open method.** |  |