Midterm “programming fundamentals

Q/ Advantages of user-defined functions.

1. Abstraction: User-defined functions help to decompose a large program into small segments which makes program easy to understand, maintain and debug.

2. Reusability: If repeated code occurs in a program. Function can be used to include those codes and execute when needed by calling that function.

3. Organizing: Programmers working on large project can divide the workload by making different functions.

Q/ Advantages of using Recursive Function:

1. Complicated function can be split down into smaller sub-problems.

2. Sequence creation is simpler through recursion than utilizing any nested iteration.

3. Function code looks simple and effective.

Q/ Disadvantages of using Recursive Function:

1. a lot of memory and time is taken through recursive calls.

2. Recursive functions are challenging to debug

ASCII/UNICODE

Decimal	Char	Decimal	Char	Decimal	Char
32	[SPACE]	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	|
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_	127	[DEL]

Not*

- **If you want to know a specific character's ASCII/UNICODE code point value, you can use a function named ord().**
- **If you know the code point (number) and want to get the corresponding character, you can use a function named chr().**

Method of string:

Method	Description
capitalize()	Converts the first character to upper

Method	Description
append()	Add an element to the end of the list
extend()	Add all elements of a list to the another list
insert()	Insert an item at the defined index
remove()	Removes an item from the list
pop()	Removes and returns an element at the given index
clear()	Removes all items from the list
index()	Returns the index of the first matched item
count()	Returns the count of the number of items passed as an argument
sort()	Sort items in a list in ascending order
reverse()	Reverse the order of items in the list

Method of list:

Method of Tuple:

Method	Description
index()	Returns the index of the first matched item
count()	Returns the count of the number of items passed as an argument

The different between (String, list, Tuple, set, dictionary

String	List	Tuple	Set	Dictionary
Immutable	Mutable	Immutable	Mutable	Mutable
Ordered/ Indexed	Ordered/ Indexed	Ordered/ Indexed	Unordered	Unordered
Allows Duplicate Members	Allow Duplicate Members	Allow Duplicate Members	Doesn't allow Duplicate Members	Doesn't allow Duplicate keys
Empty string = ""	Empty list = **[ ]**	Empty tuple = ( )	Empty set = **set( )**	Empty dictionary = **{ }**
String with single element = **"H"**	List with single item = **["Hello"]**	Tuple with single item = **("Hello")**	Set with single item = **{"Hello"}**	Dictionary with single item = **{"Hello":1}**
	It can store any data types str, list, set, tuple, int and dictionary	It can store any data types str, list, set, tuple, int and dictionary.	It can store data types (int, str, tuple) but not (list, set, dictionary)	Inside of dictionary key can be int, str, and tuple only values can be o any data type in str, list, set and dictionary.

The different between Tuple and list:

Tuple	Lists
Tuple is immutable i.e. cannot be changed after assignment.	Lists are mutable i.e. can be changed.
Tuple uses Parenthesis for comma-separated values. (Optional Parenthesis)	Lists uses square brackets for comma-separated values. (Mandatory square brackets)
We cannot modify Tuples.	We can modify Lists.
Faster	Slower, compared to Tuple
Create a Tuple: mytuple = (1,25,100);	Create a List: list1 = [10, 25, 100]

Operations and Functions on Set

Operation	Mathematical Notation	Python Syntax	Result Type	Meaning
Union	$A \cup B$	`A | B`	`set`	Elements in A or B or both
Intersection	$A \cap B$	`A & B`	`set`	Elements common to both A and B
Set Difference	$A - B$	`A - B`	`set`	Elements in A but not in B
Symmetric Difference	$A \oplus B$	`A ^ B`	`set`	Elements in A or B, but not both

Method of set:

Method	Description
add()	Adds an element to the set(If the element is already present, it doesn't add any element.)
discard()	Removes an element from the set if it is a member. (Do nothing if the element is not in set)
A.issubset(B)	returns **True** if all elements of **set A** are contained in **set B.**

Method of dictionary:

Method	Description
clear()	Removes all items from the dictionary.
items()	Returns a view object that displays a list of dictionary's (key, value) tuple pairs.
keys()	Returns a view object that displays a list of all the keys in the dictionary
values()	Returns a view object that displays a list of all the values in the dictionary.

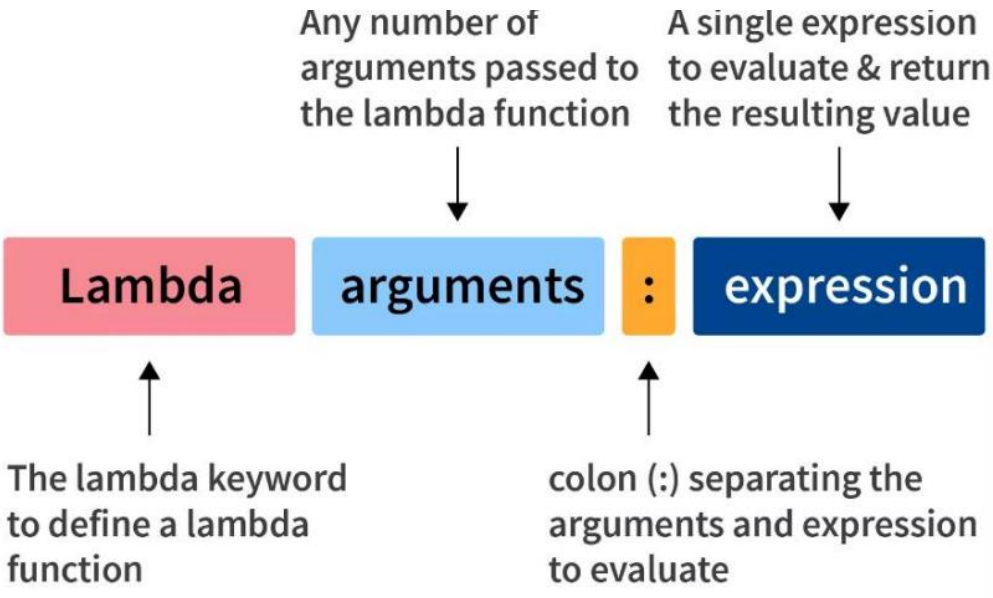
Opening Files in Python:

Character	Meaning
`'r'`	open for reading (default)
`'w'`	open for writing, truncating the file first
`'x'`	open for exclusive creation, failing if the file already exists
`'a'`	open for writing, appending to the end of the file if it exists
`'b'`	binary mode
`'t'`	text mode (default)
`'+'`	open a disk file for updating (reading and writing)

The different between Normal function and lambda function:

Normal Function	Lambda Function
Regular Function use **def** keyword	Lambda Function use **lambda** keyword
return is required in Regular Function	return is not required in Lambda Function
Return element will be python data-type	Return element will be function object
Execution time is slower	Execution time is faster

Syntax of Python Lambda



filter() Syntax

```
filter(function, iterable)
```

map() Syntax

```
map(function, iterables)
```

Python Variable Scope:

In Python, we can declare variables in three different scopes. We can classify Python variables into three types:

1. Local Variables
2. Global Variables
3. The global Keyword

Local Variable Vs. Global Variables

Comparision Basis	Global Variable	Local Variable
Definition	declared outside the functions	declared within the functions
Lifetime	They are created the execution of the program begins and are lost when the program is ended	They are created when the function starts its execution and are lost when the function ends
Data Sharing	Offers Data Sharing	It doesn't offers Data Sharing
Scope	Can be access throughout the code	Can access only inside the function
Parameters needed	parameter passing is not necessary	parameter passing is necessary
Storage	A fixed location selected by the compiler	They are kept on the stack
Value	Once the value changes it is reflected throughout the code	once changed the variable don't affect other functions of the program