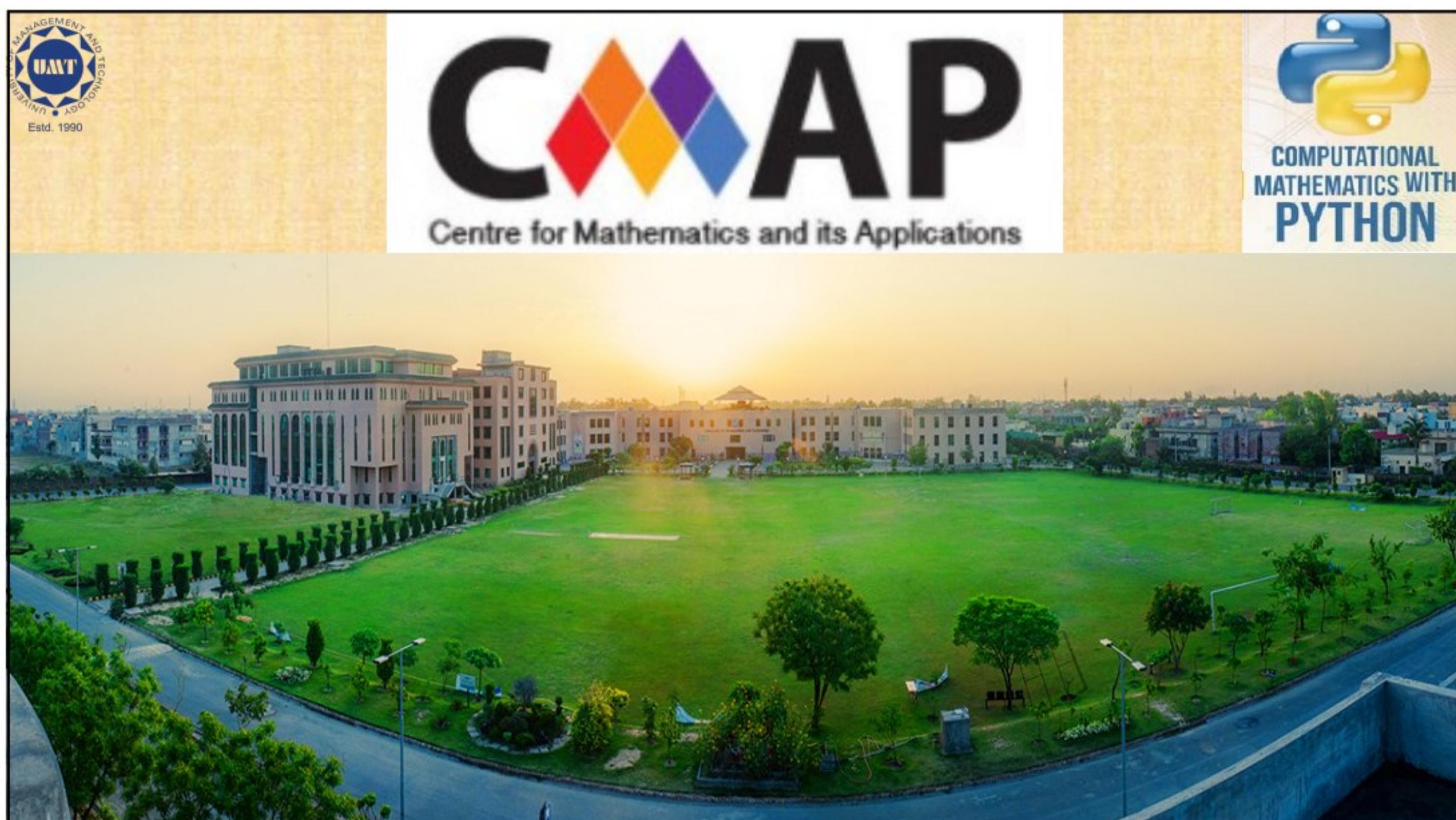
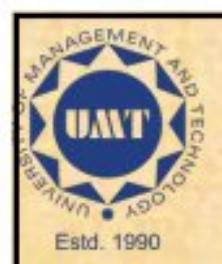


**Dr. Ghulam Murtaza** is a highly accomplished expert in Mathematical Analysis, with a decade of teaching experience, presently working at the University of Management and Technology (UMT). His diverse interests span machine learning, fuzzy mathematics, soft set theory, cryptography, fixed point theory, and mathematical inequalities. Dr. Murtaza's research contributions are well-recognized, with numerous articles published in respected mathematics journals. He is also a prominent figure in the academic and research communities at UMT and in Pakistan, known for his active participation in international conferences on computational mathematics. Dr. Murtaza's dedication to both teaching and research makes him a significant influencer in the field of Mathematics.

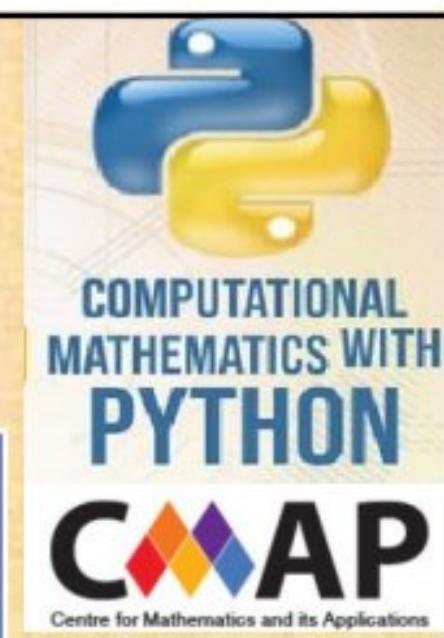
Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)





## LEARNING OBJECTIVES OF THE WORKSHOP

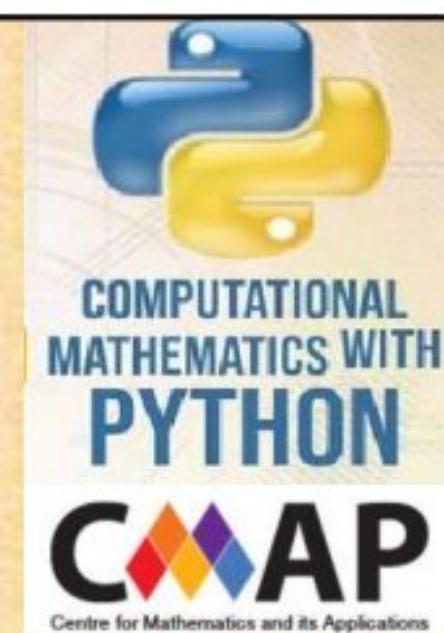
- Installation and basic programming structures of Python
- Numpy and basics of graphing using Matplotlib
- Matrices operations, derivatives and integration
- Ordinary differential equations



Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



## LECTURE 1

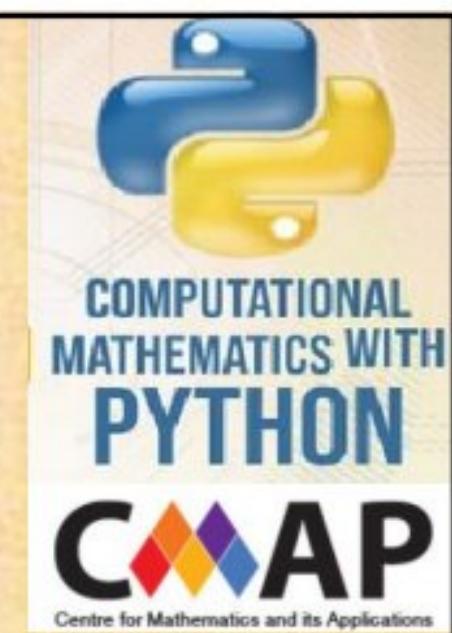


Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

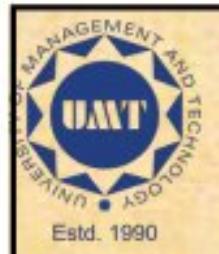


## LEARNING OBJECTIVES

- Installing Jupyter
- Getting familiar with Jupyter notebook
- Getting familiar with Google Colab
- Python programming
  - Basic data types,
  - Containers(Lists, Dictionaries, Sets, Tuples),
  - Conditional statements,
  - Functions,
  - Classes

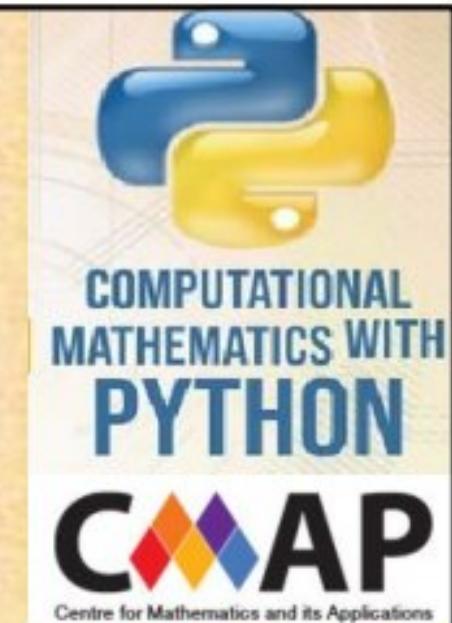


Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



## PYTHON NOTEBOOKS

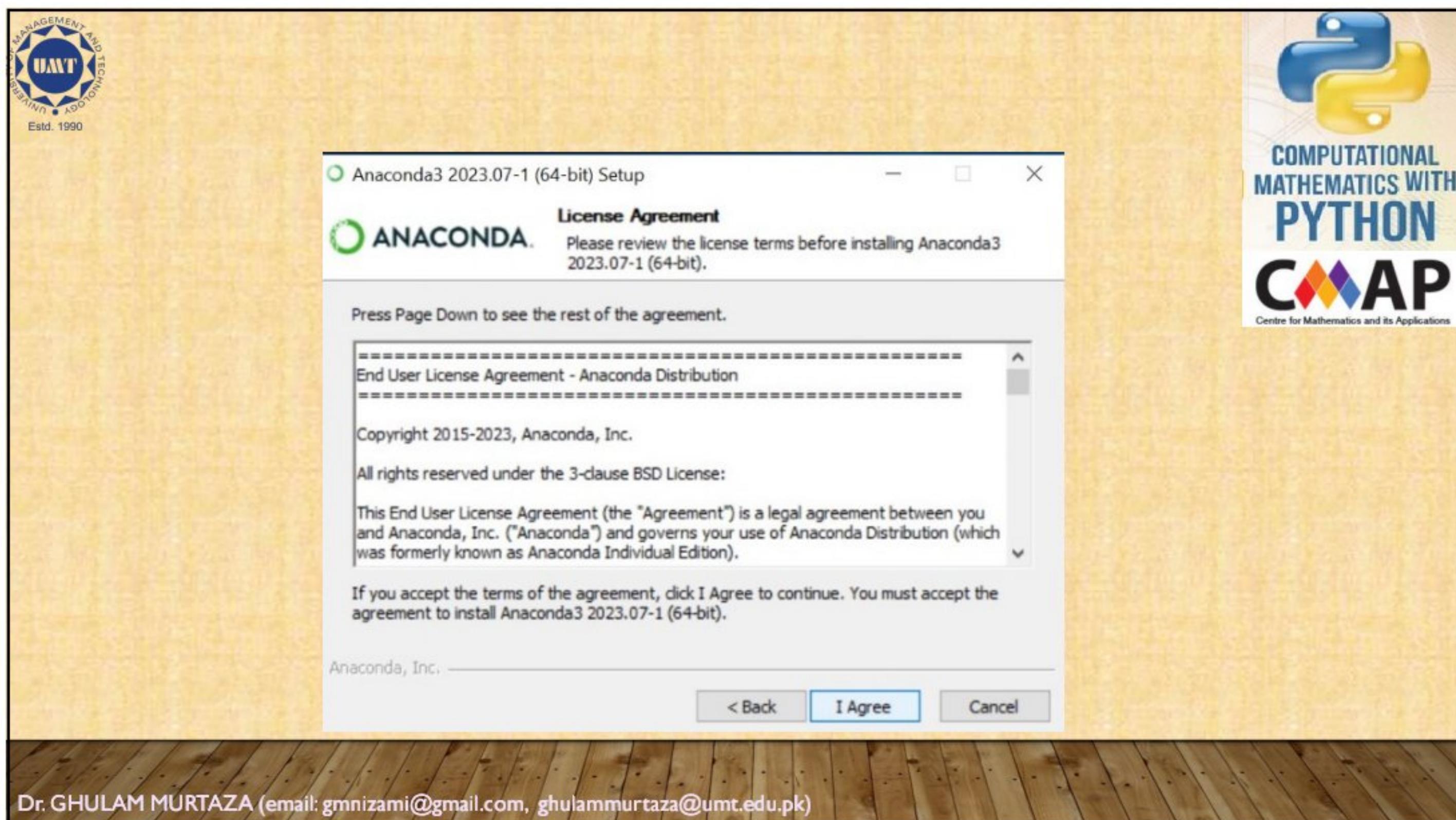
- Jupyter 



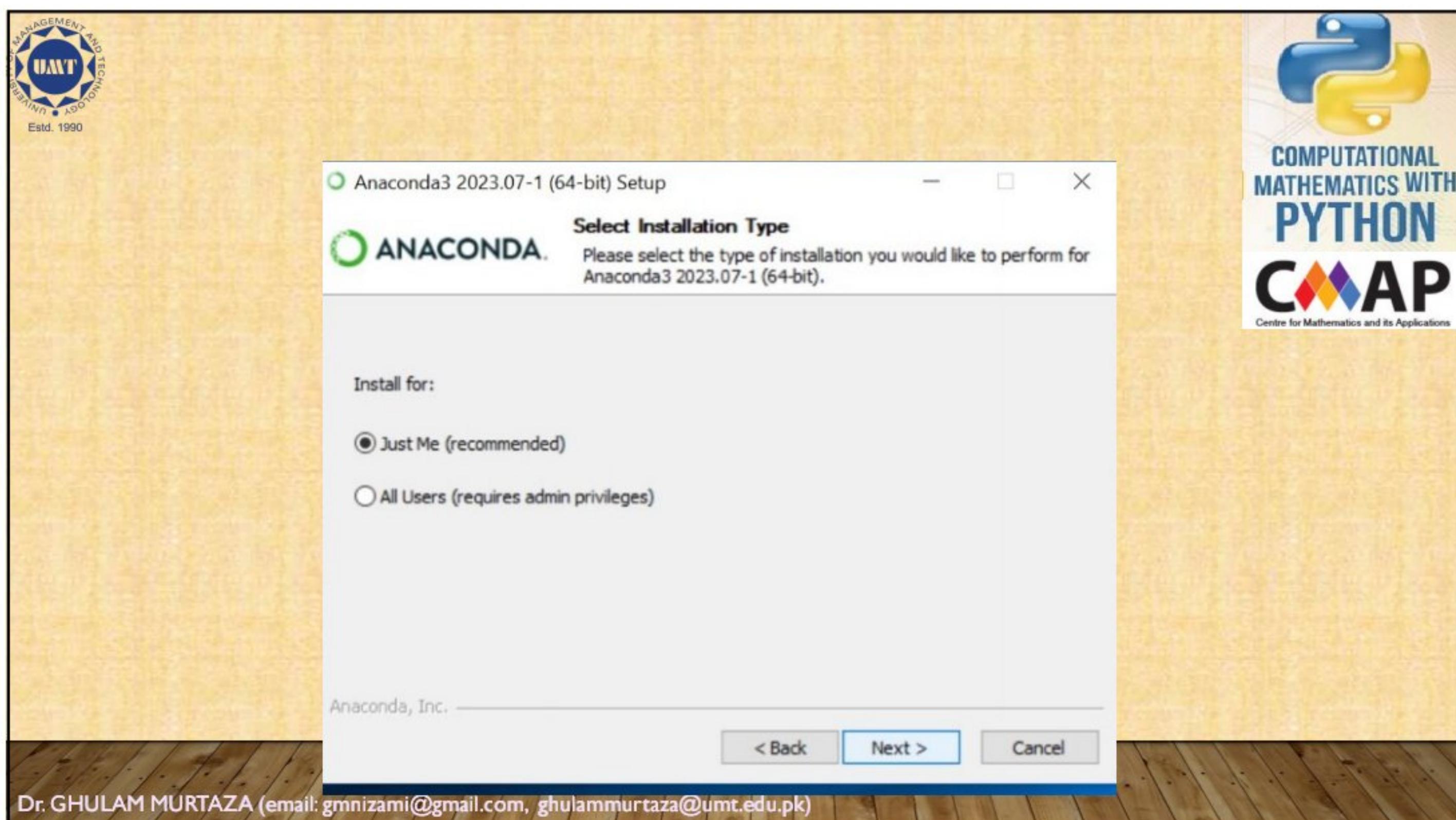
- Google Colab 

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

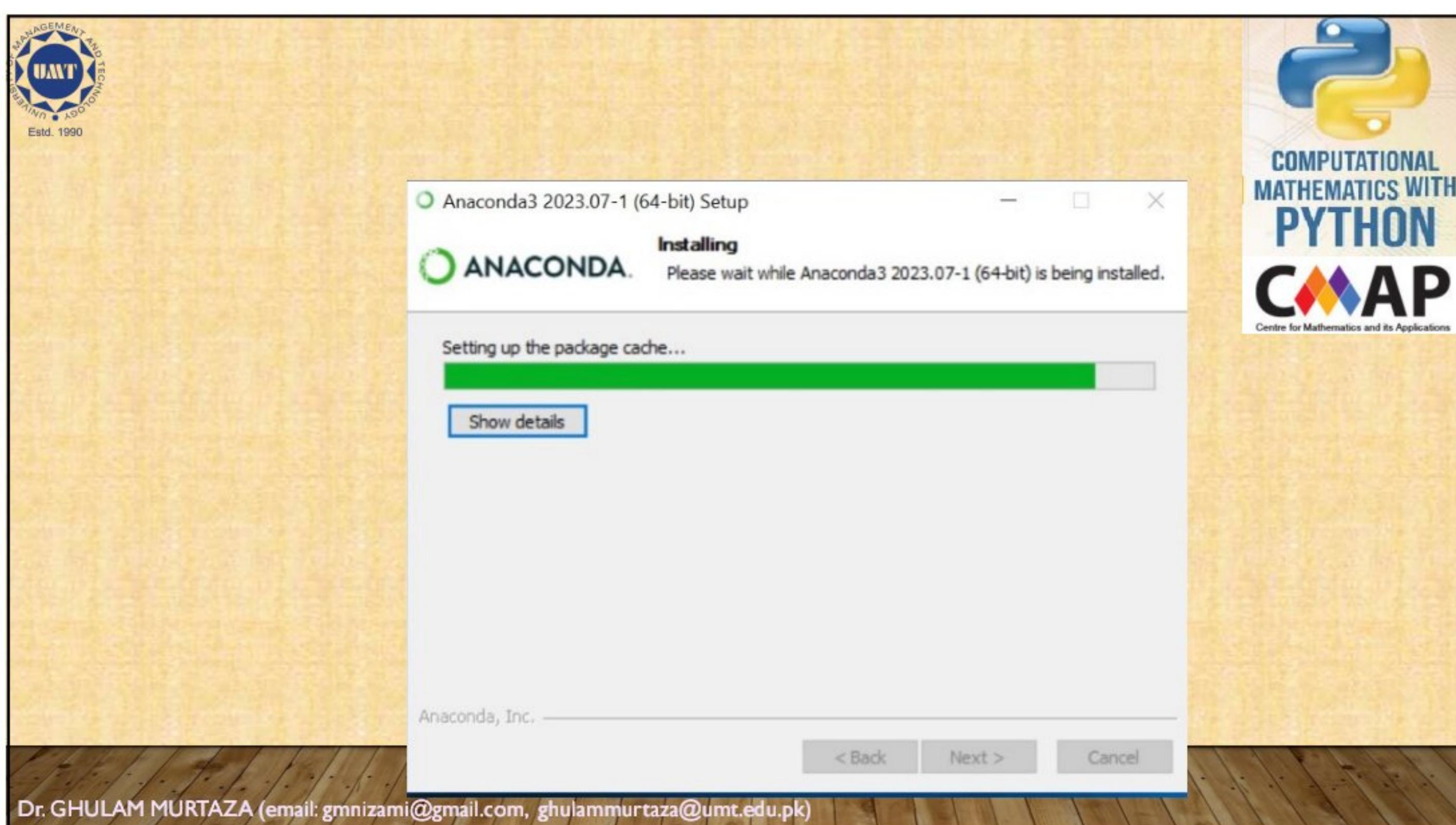
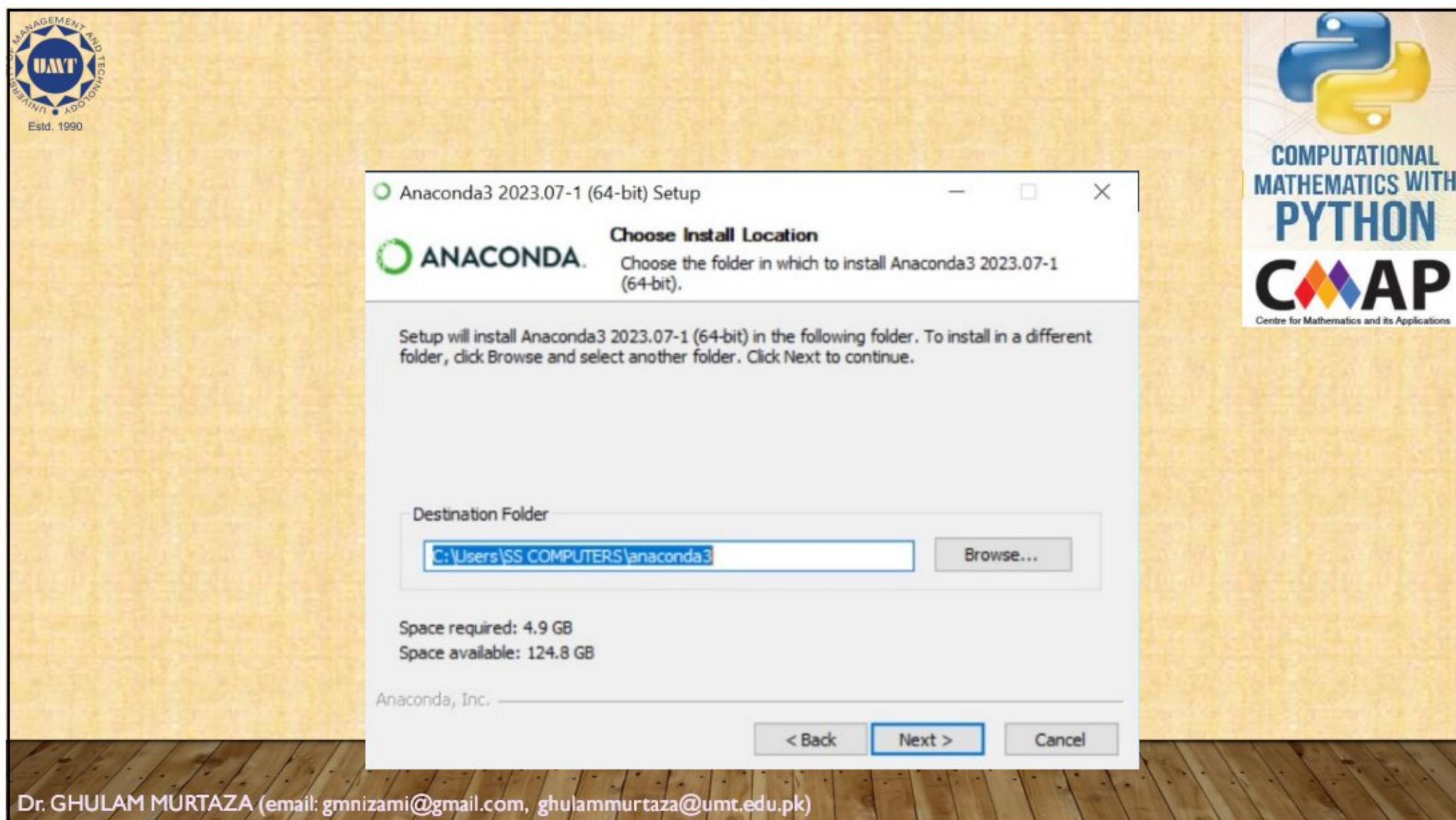


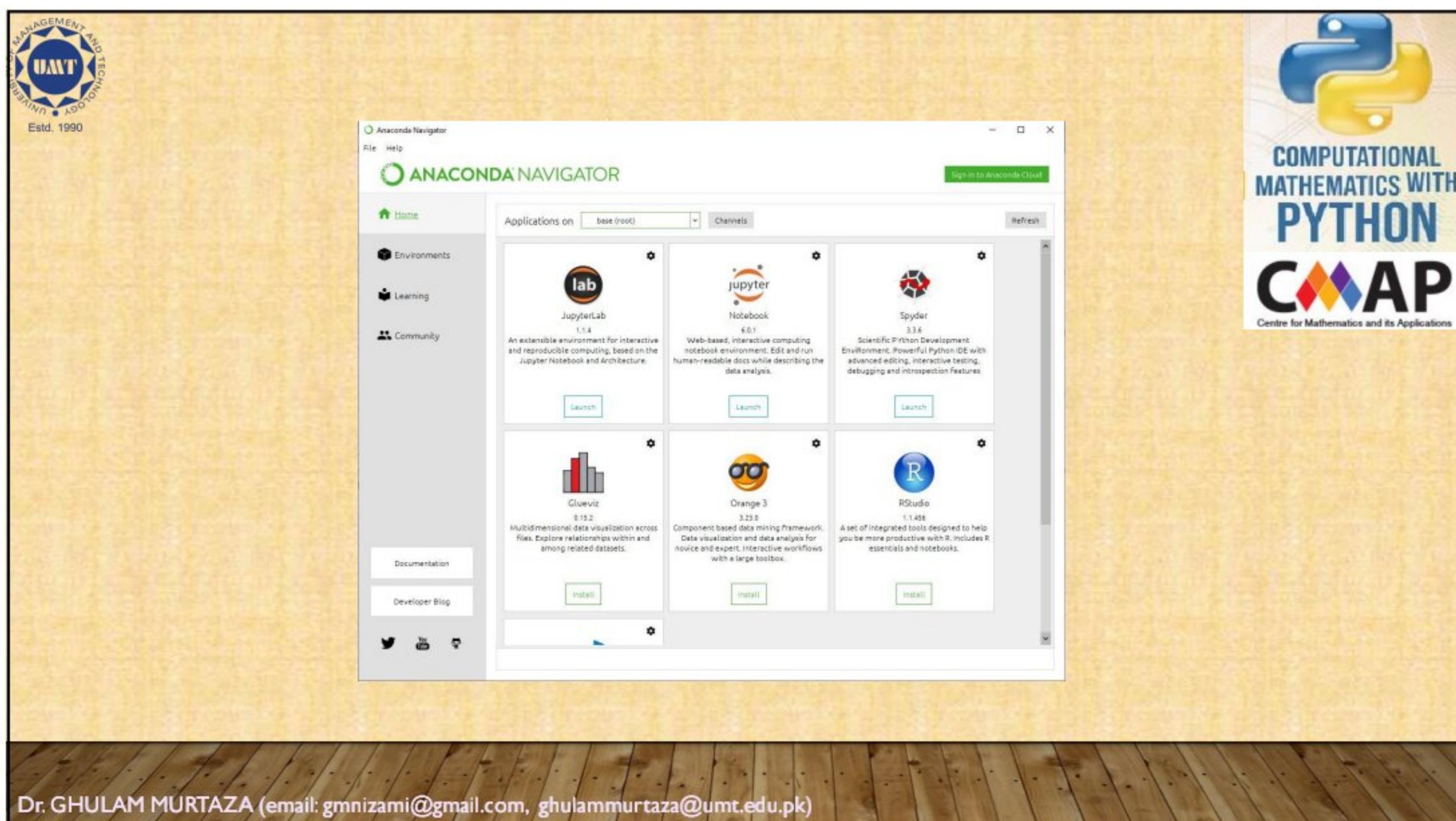


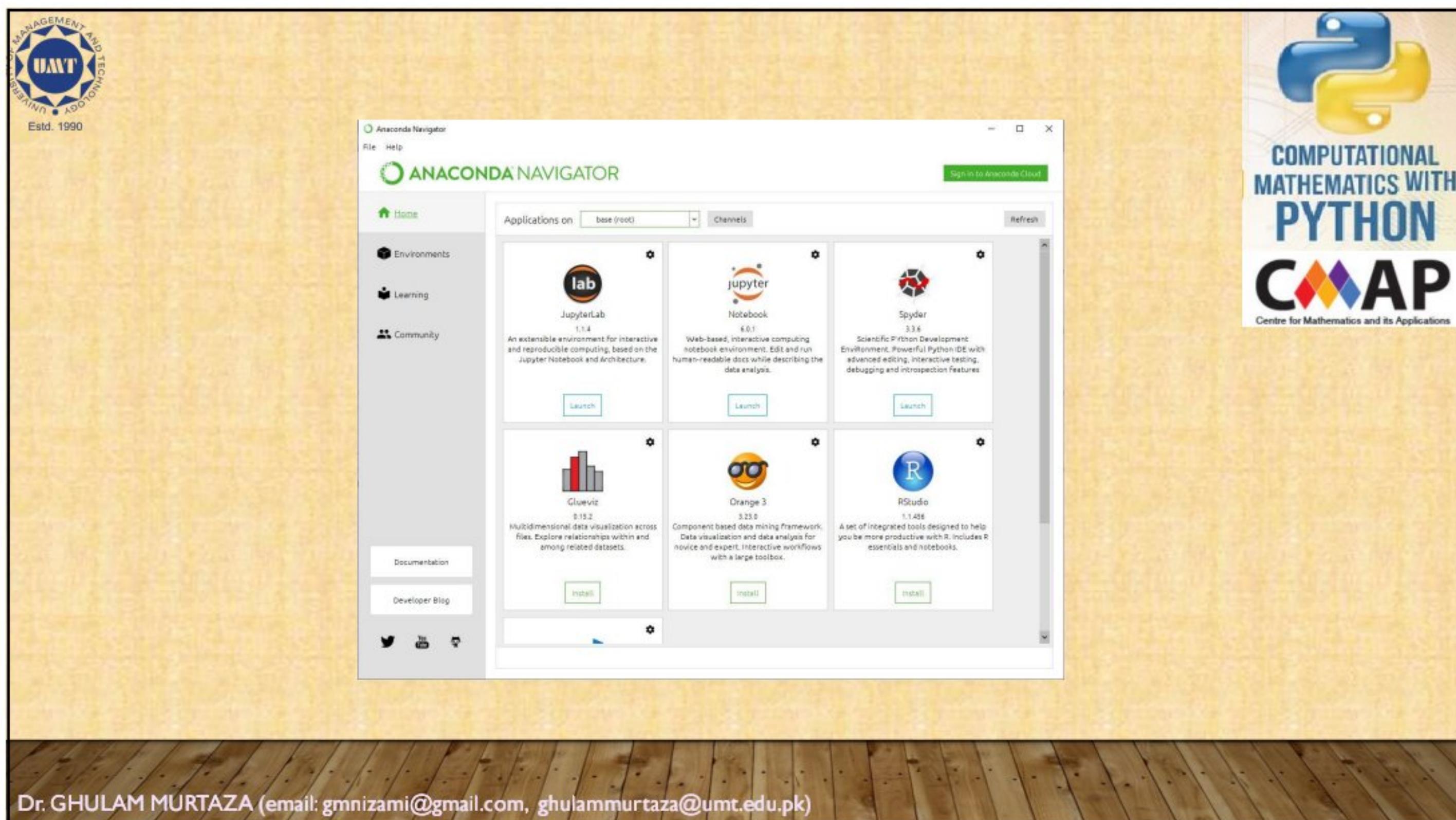
Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



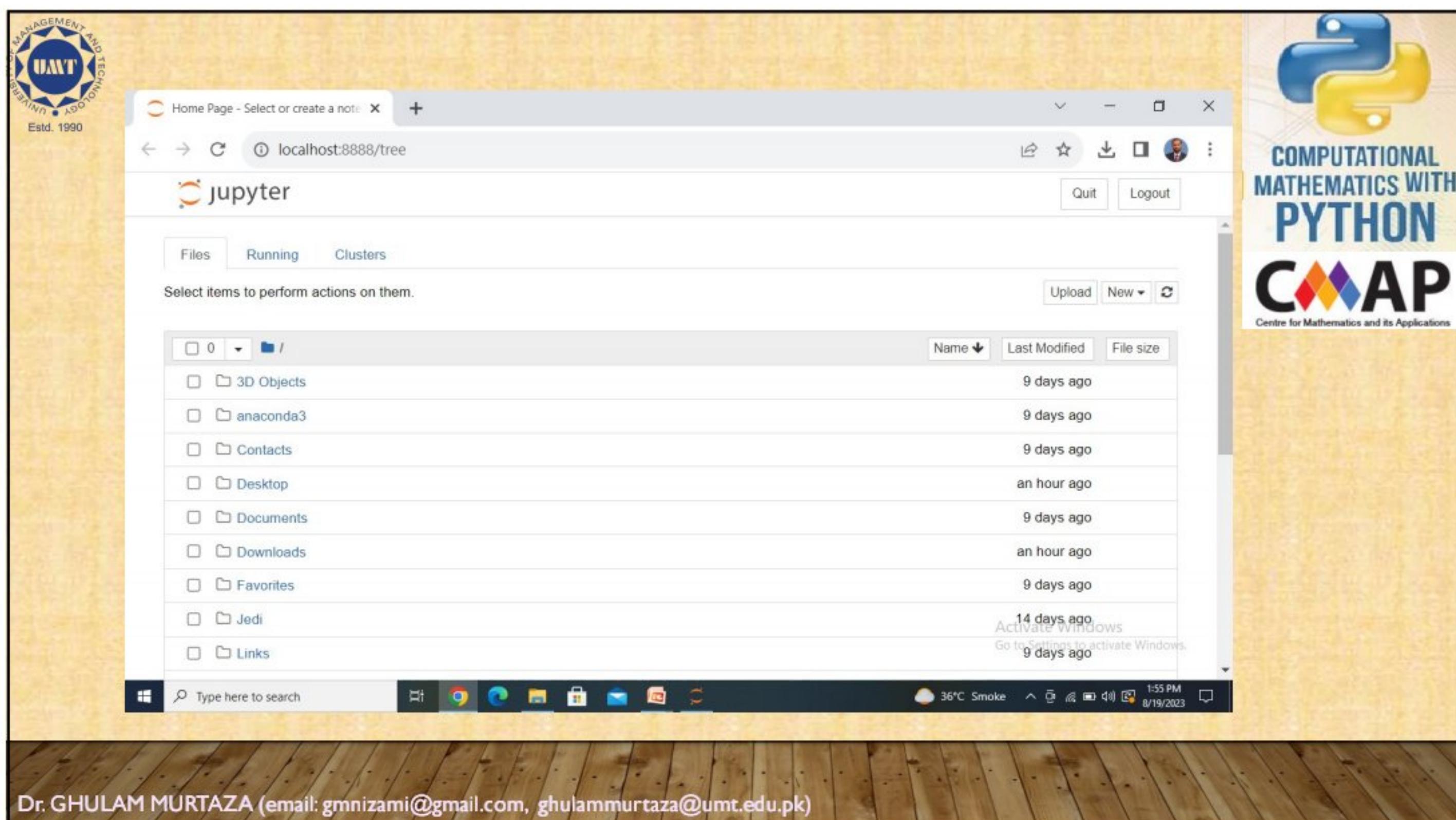
Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



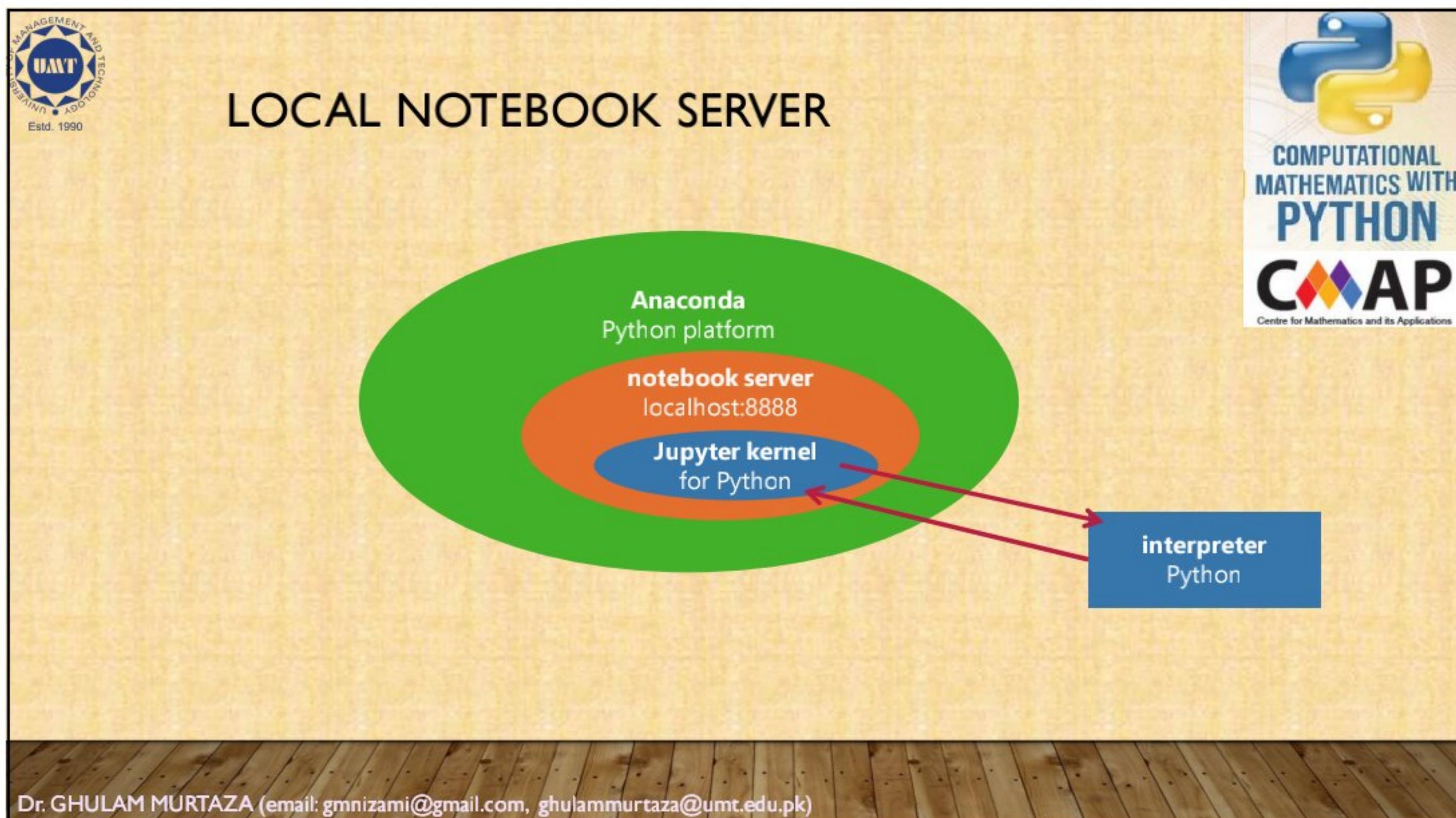




Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



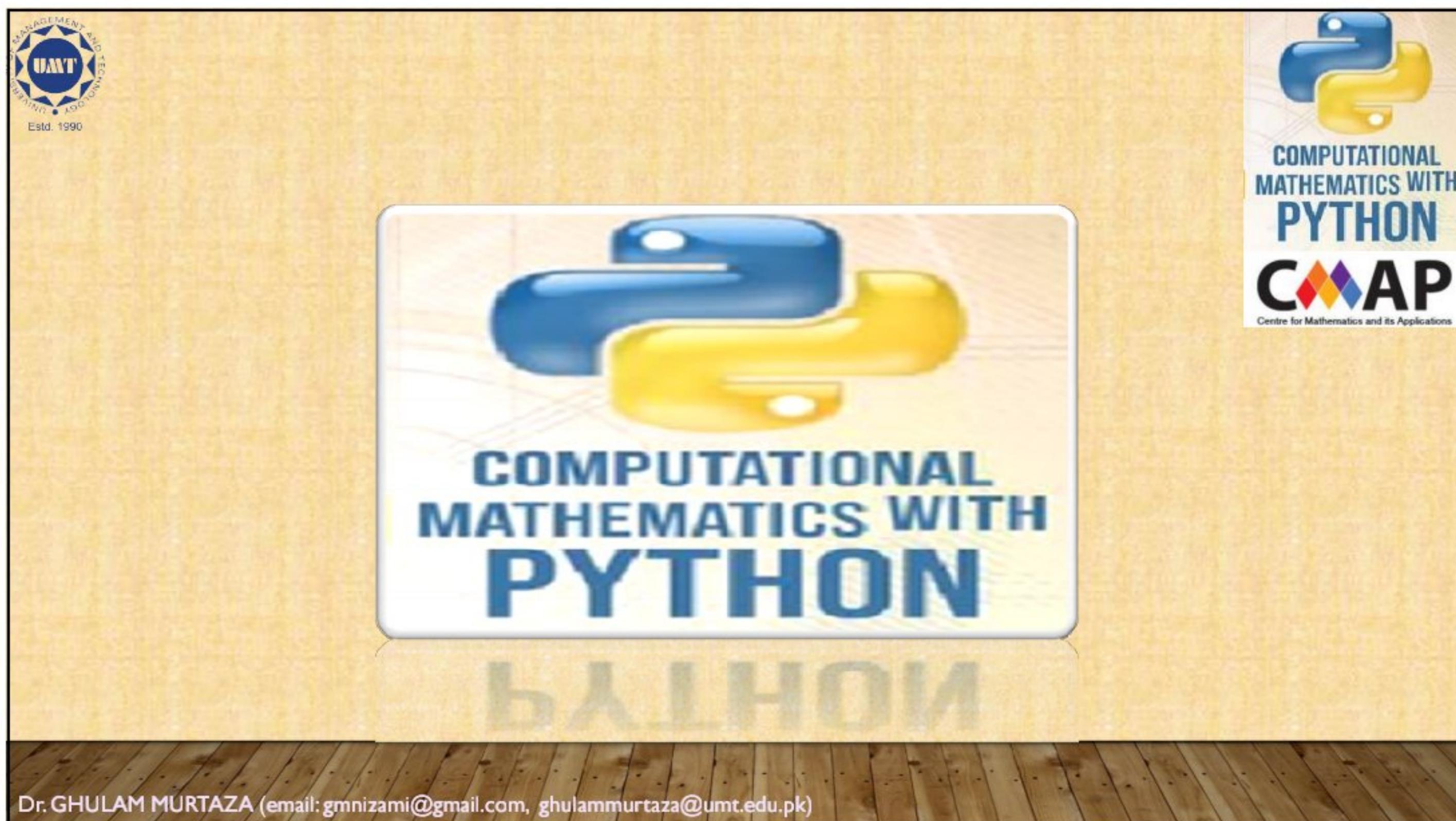
**INSTALLING PACKAGES**

- You can install packages from the terminal using the command:  
`>>conda install <name of package>`  
Or  
`>>pip install <name of package>`
- You can install packages from writing the following code in code cell of Jupyter:  
`!pip install <name of package>`
- For example, Fuzzysets is a package for fuzzy sets.  
`!pip install fuzzysets`
- For uninstall run the code:** `!pip uninstall --y <name of package>`

**Conda** is an open source package management system

The Python Package Index (**PyPI**) is a repository of software for the Python programming language

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



A screenshot of a Jupyter Notebook interface. The window title is 'Untitled5 - Jupyter Notebook'. The notebook content shows a single code cell starting with 'In [ ]:'. A red arrow points from the text 'Cell' to the cell itself. Another red arrow points from the text 'Name of file' to the file path in the browser's address bar: 'localhost:8888/notebooks/Untitled5.ipynb?kernel\_name=python3'. A third red arrow points from the text 'Type of cell' to the word 'Code' in the toolbar above the cell.

**☐ The notebook is organized in cells**  
**☐ In each cell you can write either code or text**  
**☐ The default behavior is code**

At the bottom left, there is contact information: 'Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)'

In [1]: `print("hello world")`

hello world

In [ ]:

- You can run the code using the Run button or with Ctrl+Enter
- Note that now we have both the code and the output in the notebook

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

**## Introduction**

Python is a great general-purpose programming language on its own, but with the help of a few popular libraries (numpy, scipy, sympy, matplotlib) it becomes a powerful environment for scientific computing.

In this lecture, we will cover:

- \* Basic data types,
- \* Containers,
- \* Lists,
- \* Dictionaries,
- \* Sets,
- \* Tuples,
- \* Functions,
- \* Classes

You can also write text in Markdown language  
You can combine HTML, and Latex, and there are some other commands

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

**ATTENTION!**

- You can jump between cells in a non-linear way
- You should be aware of the state of the memory of the notebook when you run a specific cell.

jupyter Untitled5 Last Checkpoint: an hour ago (unsaved changes)

In [1]: `x=2`

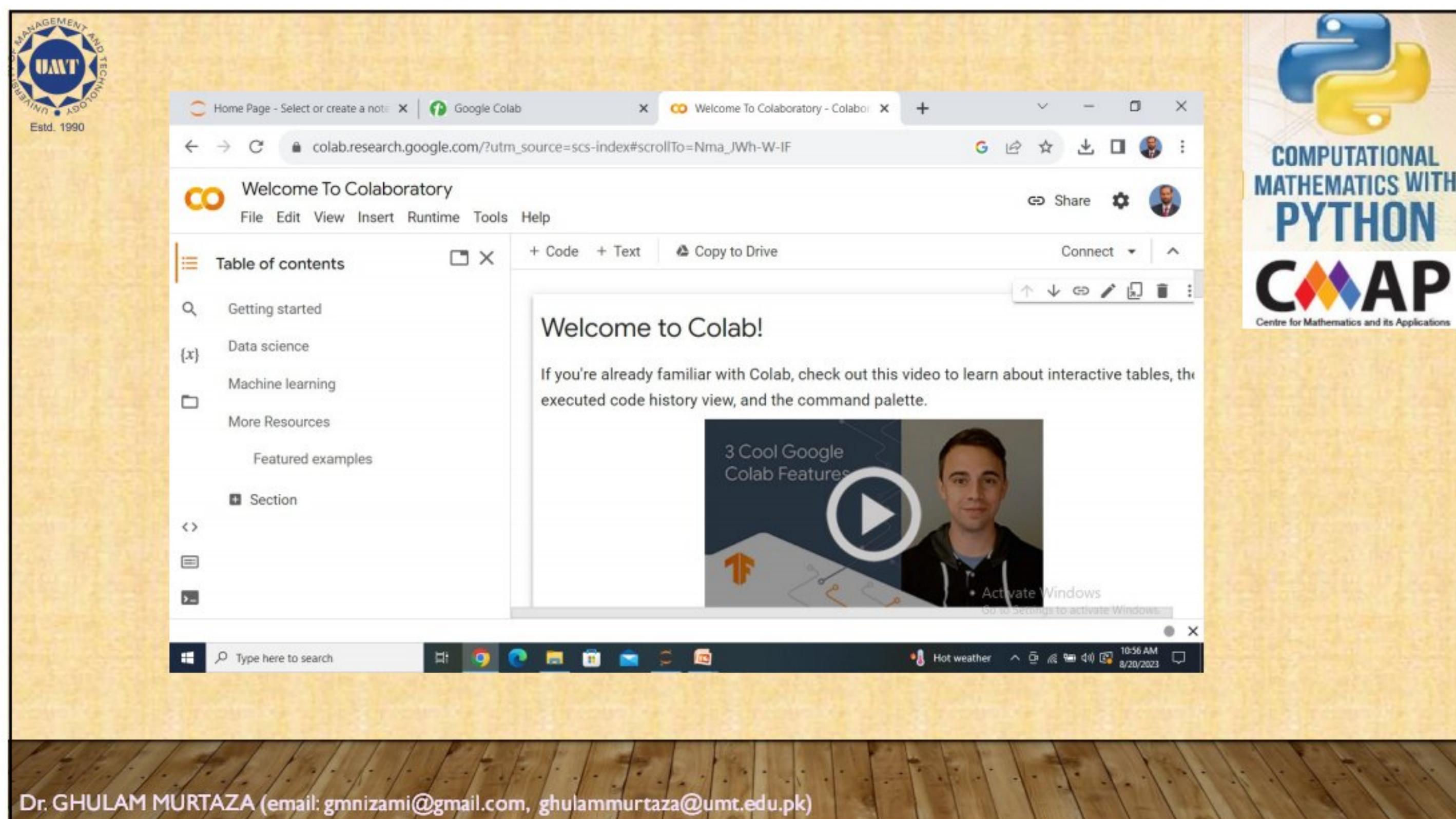
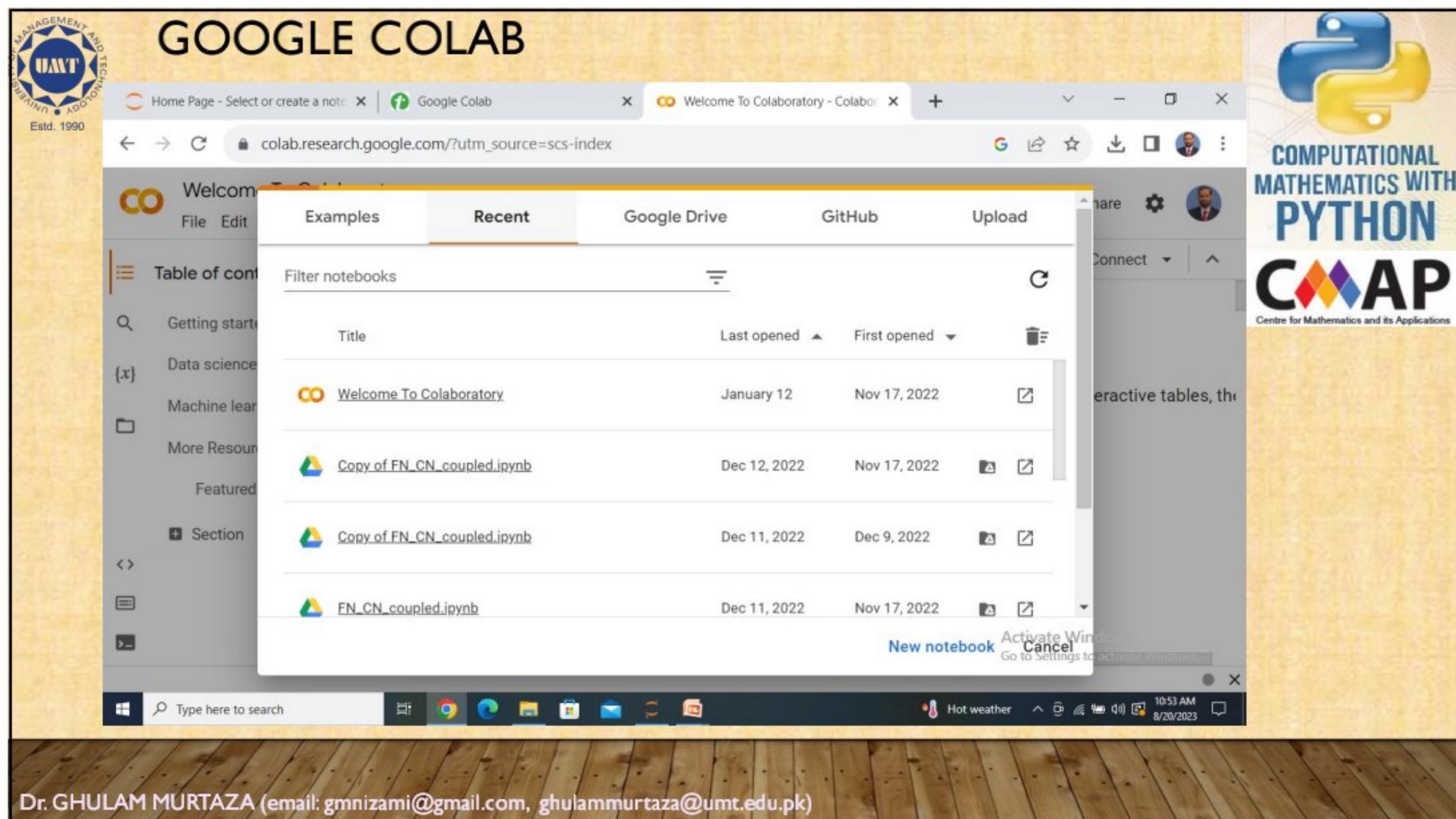
In [3]: `print("x=",x)`  
x= 5

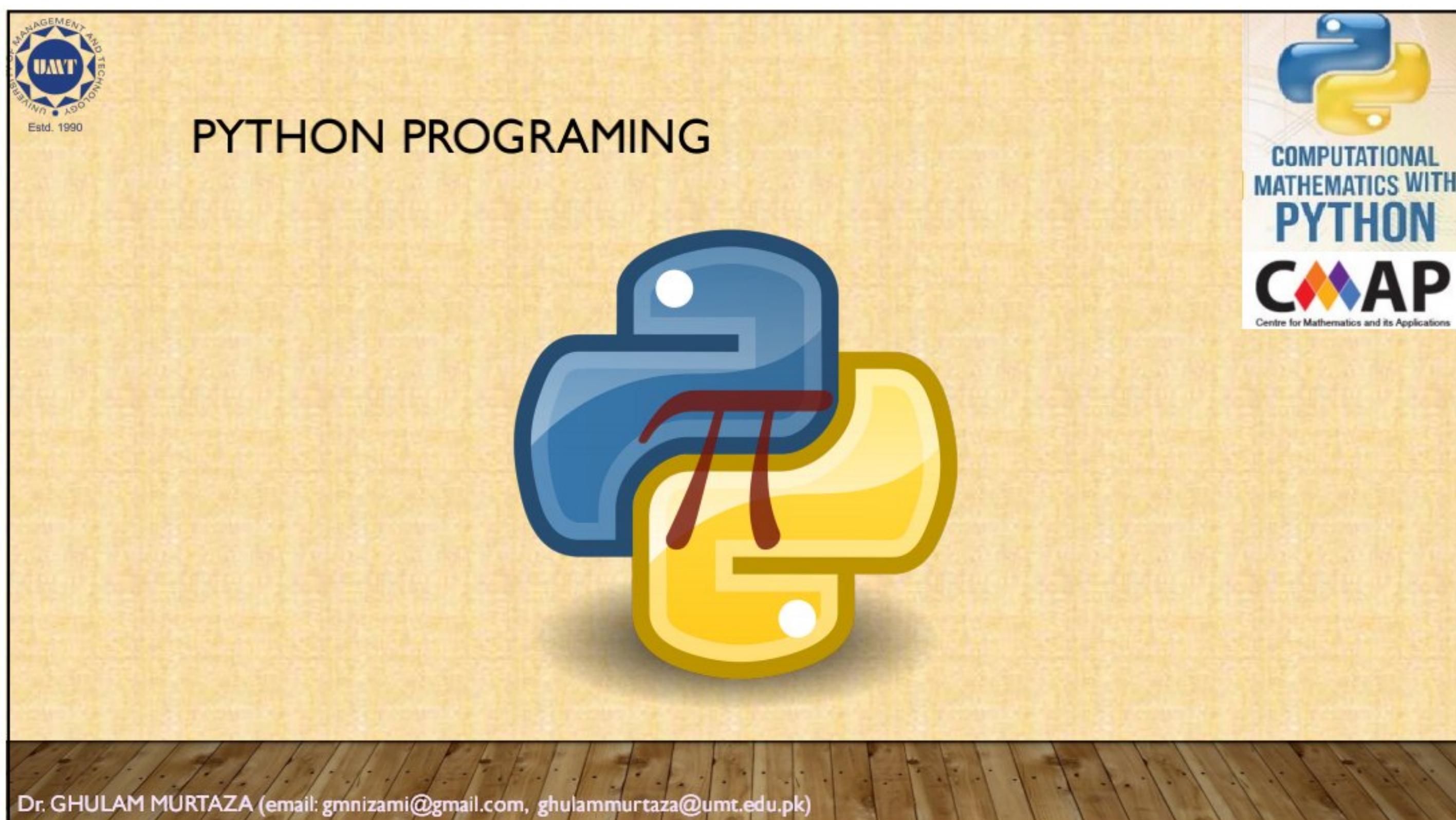
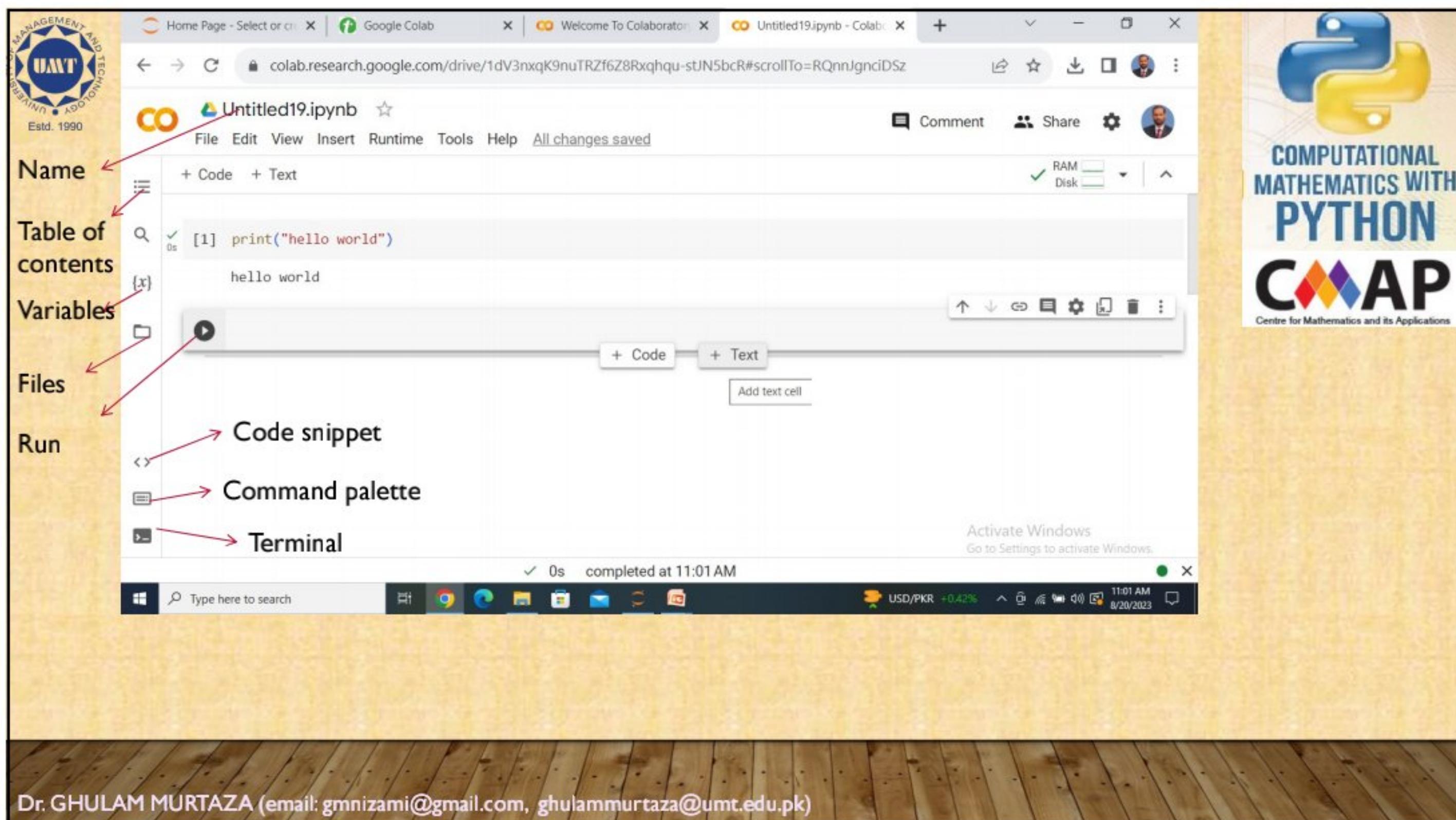
In [2]: `x=5`

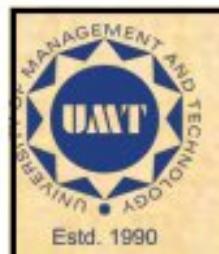
Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

COMPUTATIONAL  
MATHEMATICS WITH  
**PYTHON**

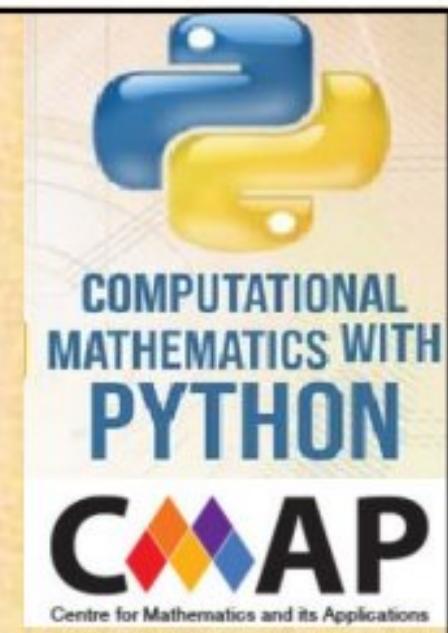
Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)







## NAMING OBJECTS



Python names must follow these rules:

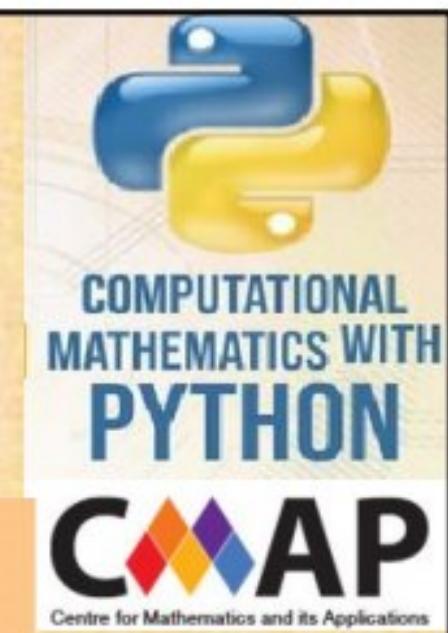
- Start with a letter (uppercase or lowercase) or an underscore (\_)
- Can contain letters (uppercase or lowercase), underscores (\_), or digits
- Cannot be Python's keywords
- Are case-sensitive

Note: Python keywords cannot be used for variable name([keywords](#))

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

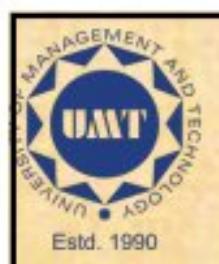


## THE “print” FUNCTION



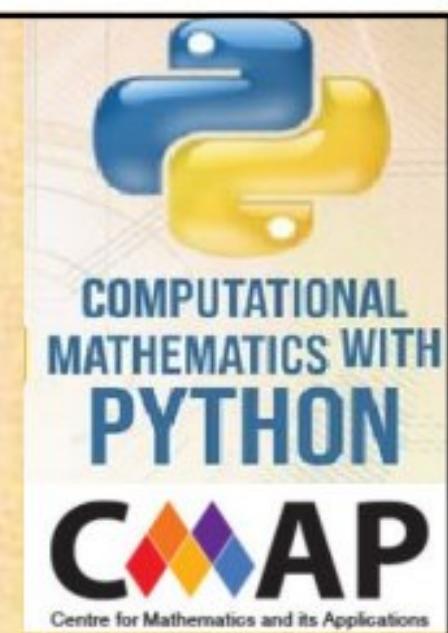
Function	Description
<code>print(param1, ... )</code>	Outputs each parameter value on the same line, separated by a space

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



## THE “input” FUNCTION

`input(prompt)`

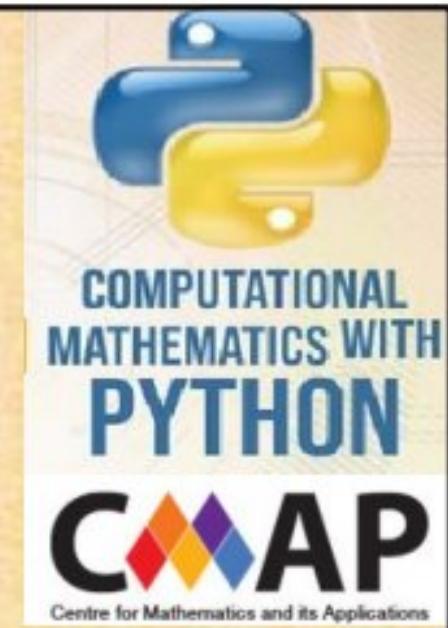
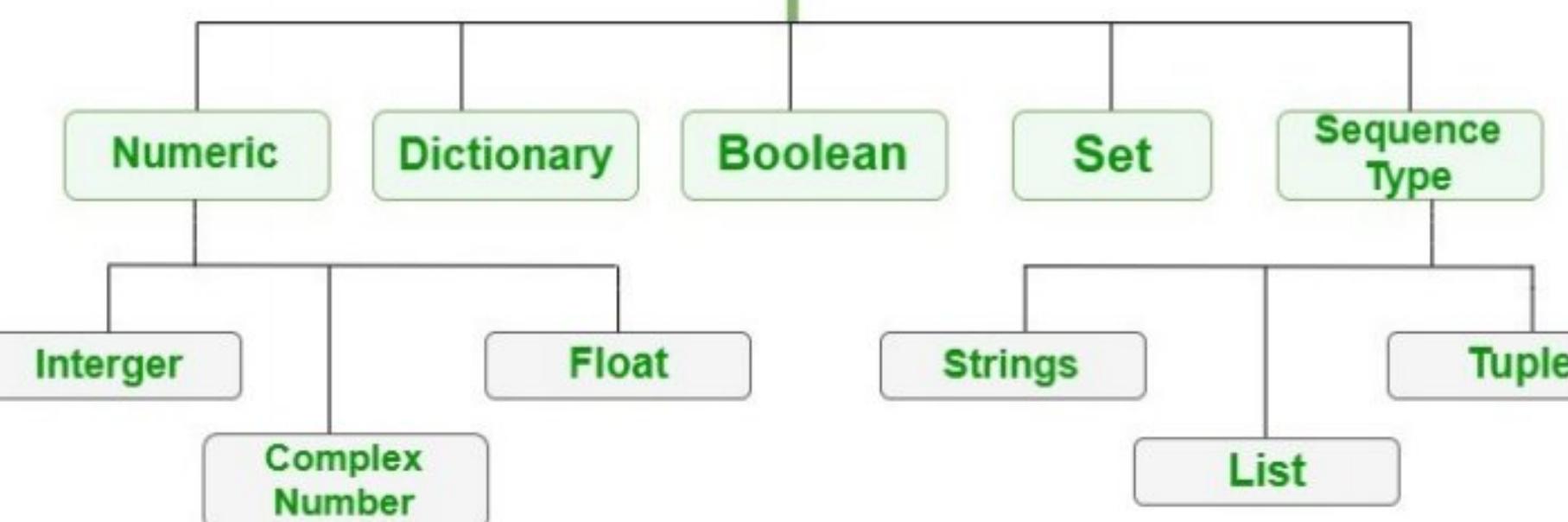


Reads a line from input, convert it into string and return that.

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



### Python - Data Types



Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

DG

 UNIVERSITY OF MANAGEMENT AND TECHNOLOGY  
ESTD. 1990

## NUMERIC OPERATORS

Operator	Operation
+	Addition
-	Subtraction
*	Multiplication
//	Integer division
/	Floating-point division
%	Modulo (remainder after division)
**	Exponentiation

  
**COMPUTATIONAL MATHEMATICS WITH PYTHON**  
**CMAP**  
Centre for Mathematics and its Applications

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

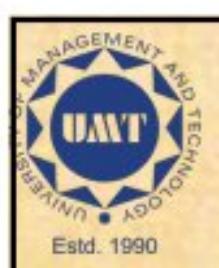
 UNIVERSITY OF MANAGEMENT AND TECHNOLOGY  
ESTD. 1990



**COMPUTATIONAL MATHEMATICS WITH PYTHON**

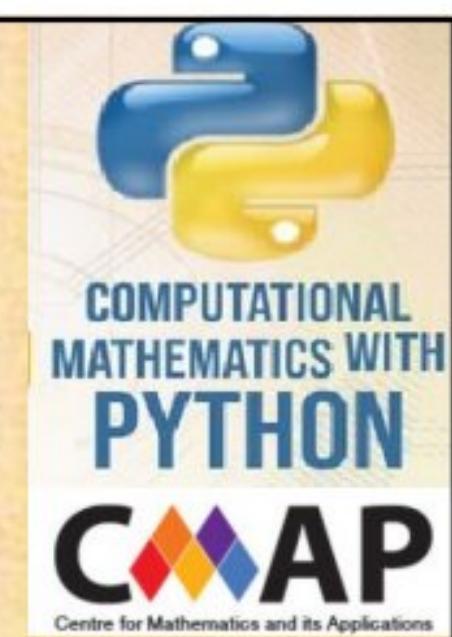
  
**COMPUTATIONAL MATHEMATICS WITH PYTHON**  
**CMAP**  
Centre for Mathematics and its Applications

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

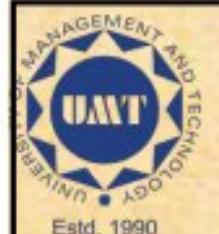


## PYTHON CONTAINERS

- Lists,
- Dictionaries,
- Sets,
- Tuples

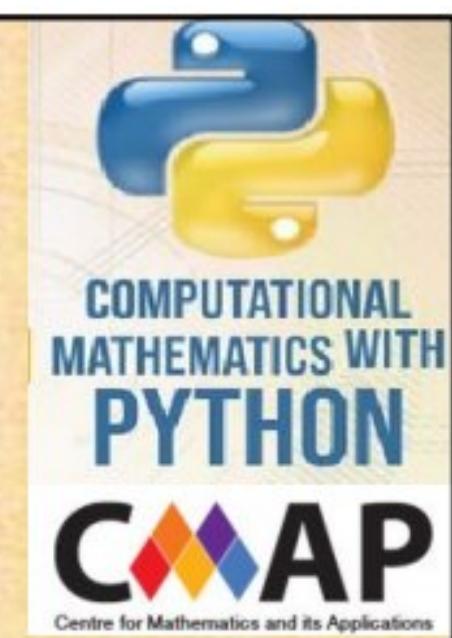


Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



### LIST

Lists are used for preserving a sequence of data (data may be of different datatype) and are ordered and mutable (it can be changed after being created).



Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

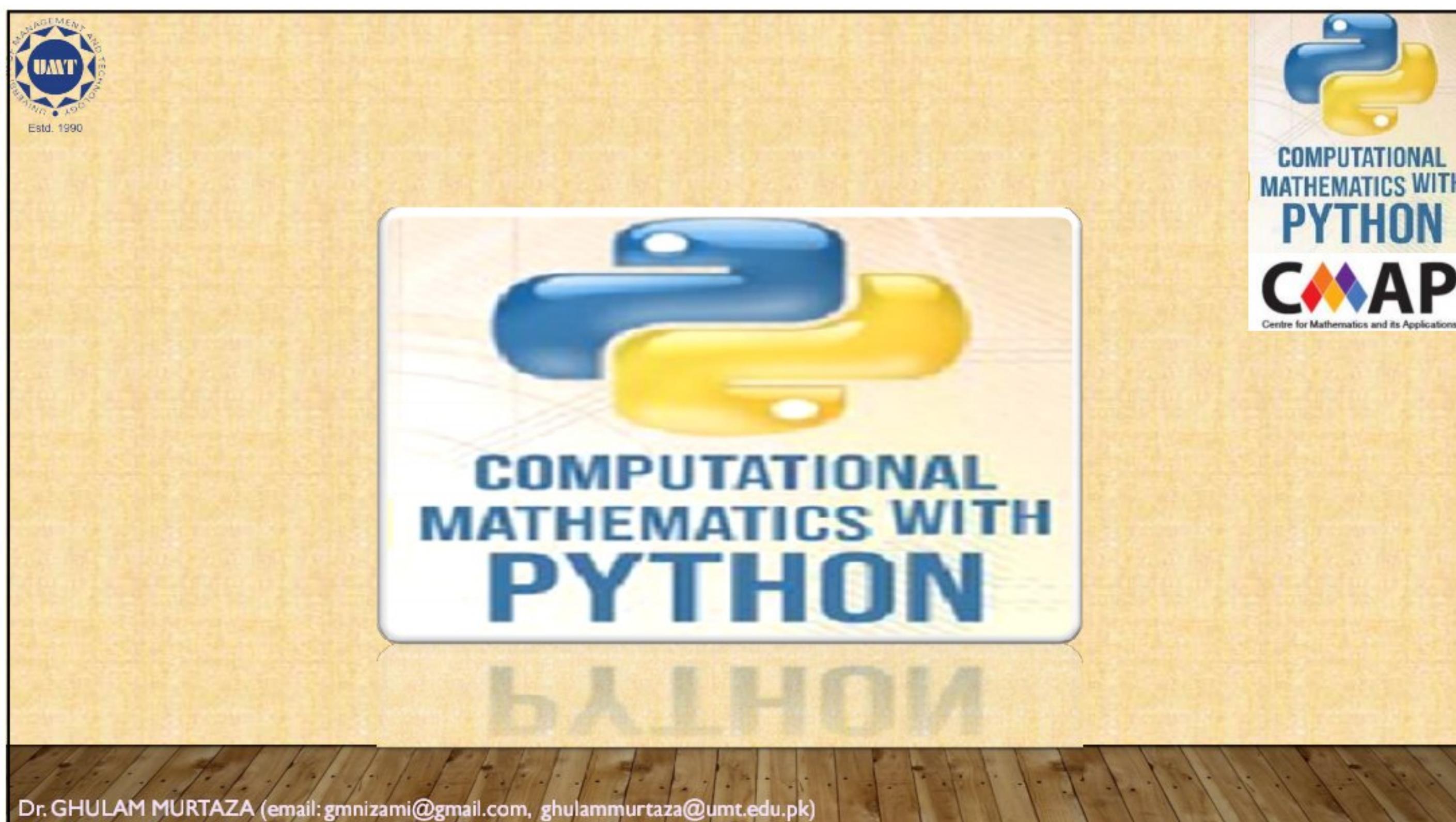


Function	To do	Example
len()	Returns number of items in the list	len(mylist)
append()	Append an item in the list	mylist.append('item3')
insert()	Insert an item in the list	mylist.insert(0,'item0')
clear()	Empty the list	mylist.clear()
del	Delete the list Delete item(s) from the specified index	del mylist, del mylist[0]
pop()	Remove item from last(or specified index)	mylist.pop(), mylist.pop(0)
index()	Returns the index of the value specified	mylist.index('item1')
count()	Returns the number of occurrence of the specified item in the list	mylist.count('item2')
sort()	Sort the list in ascending order and return None.	mylist.sort()
reverse()	Sort in reverse order	mylist.reverse()

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



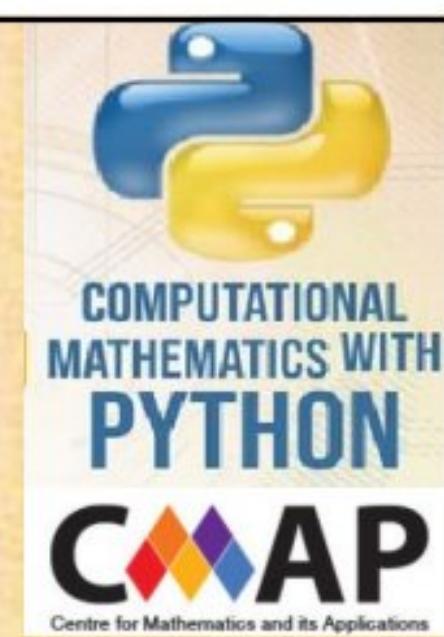
COMPUTATIONAL MATHEMATICS WITH PYTHON  
**CMAP**  
Centre for Mathematics and its Applications





## Dictionary

Dictionaries are used to store data values in key:value pairs. A dictionary is a collection which is ordered\*, changeable and do not allow duplicates.

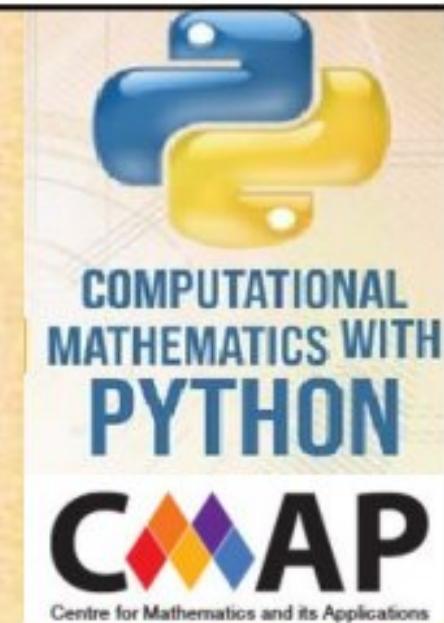


Python allows the *values* in a dictionary to be any type – string, integer, a list, another dictionary, boolean, etc. However, keys must always be an immutable data type, such as strings, numbers, or tuples.

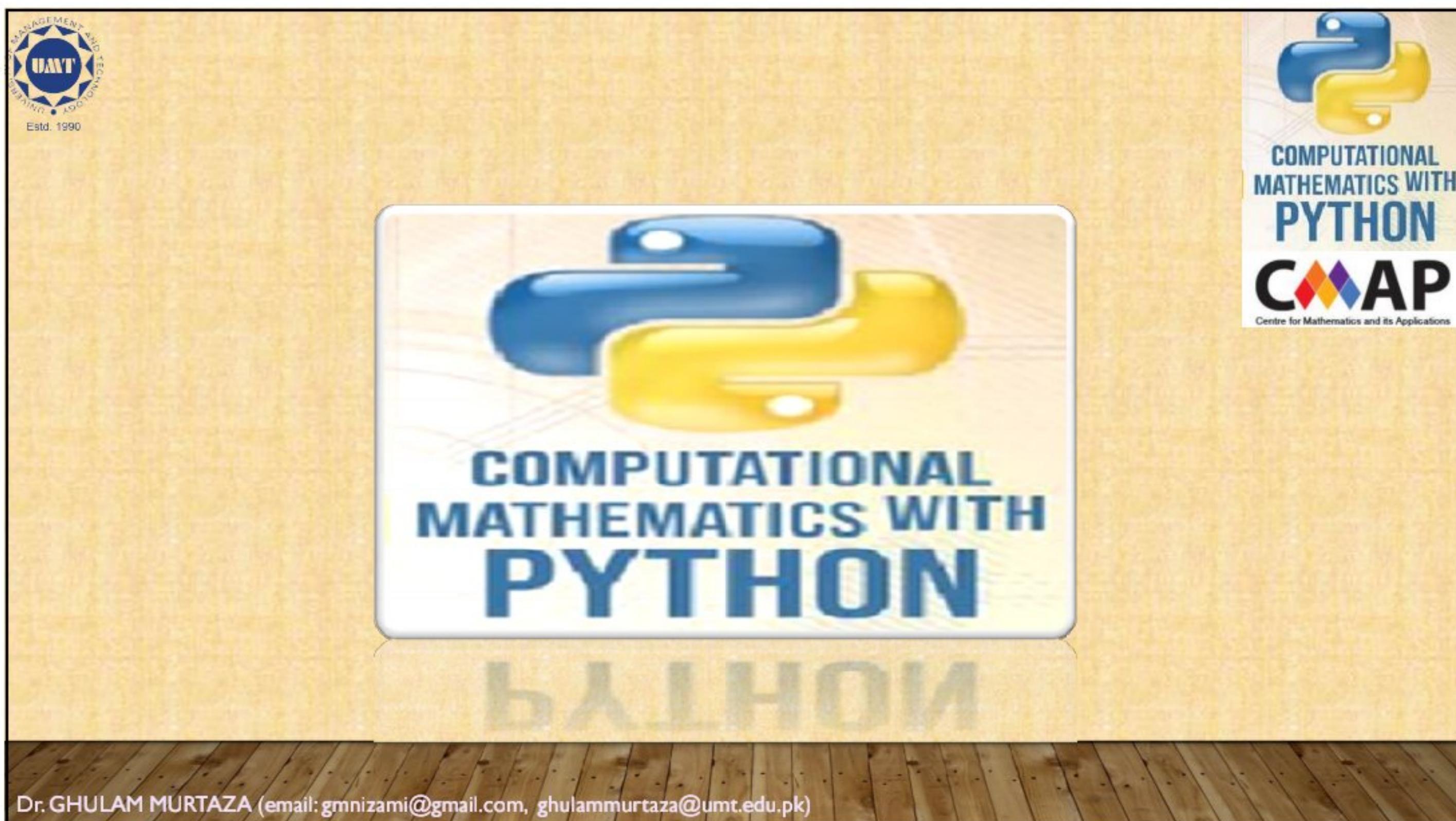
Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



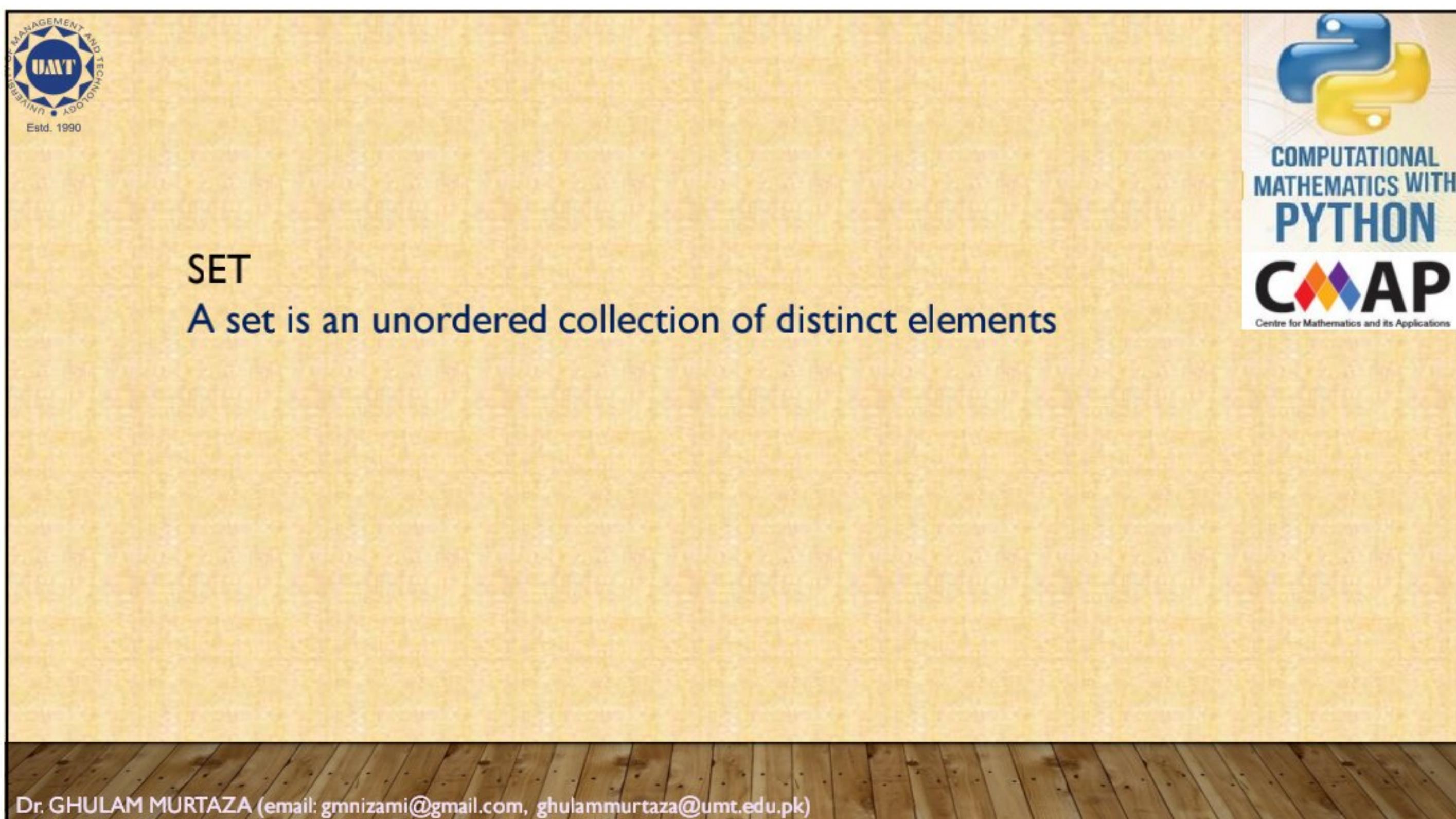
Function	To do	Example
len()	Returns number of items in the dictionary	len(mydic)
Adding a value in dictionary using key		mydic[key]=value
clear()	Empty the dictionary	mydic.clear()
del	Delete the dictionary Delete item(s) from the specified key	del mydic, del mydic[key]
pop()	Remove item from dictionary for specified key	mydic.pop(key),
keys()	Returns the all keys of the dictionary	mydic.keys()
values()	Returns the all values in the dictionary	mydic.values()
items()	Returns the list with all dictionary keys with values as tuples.	mydic.items()



Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

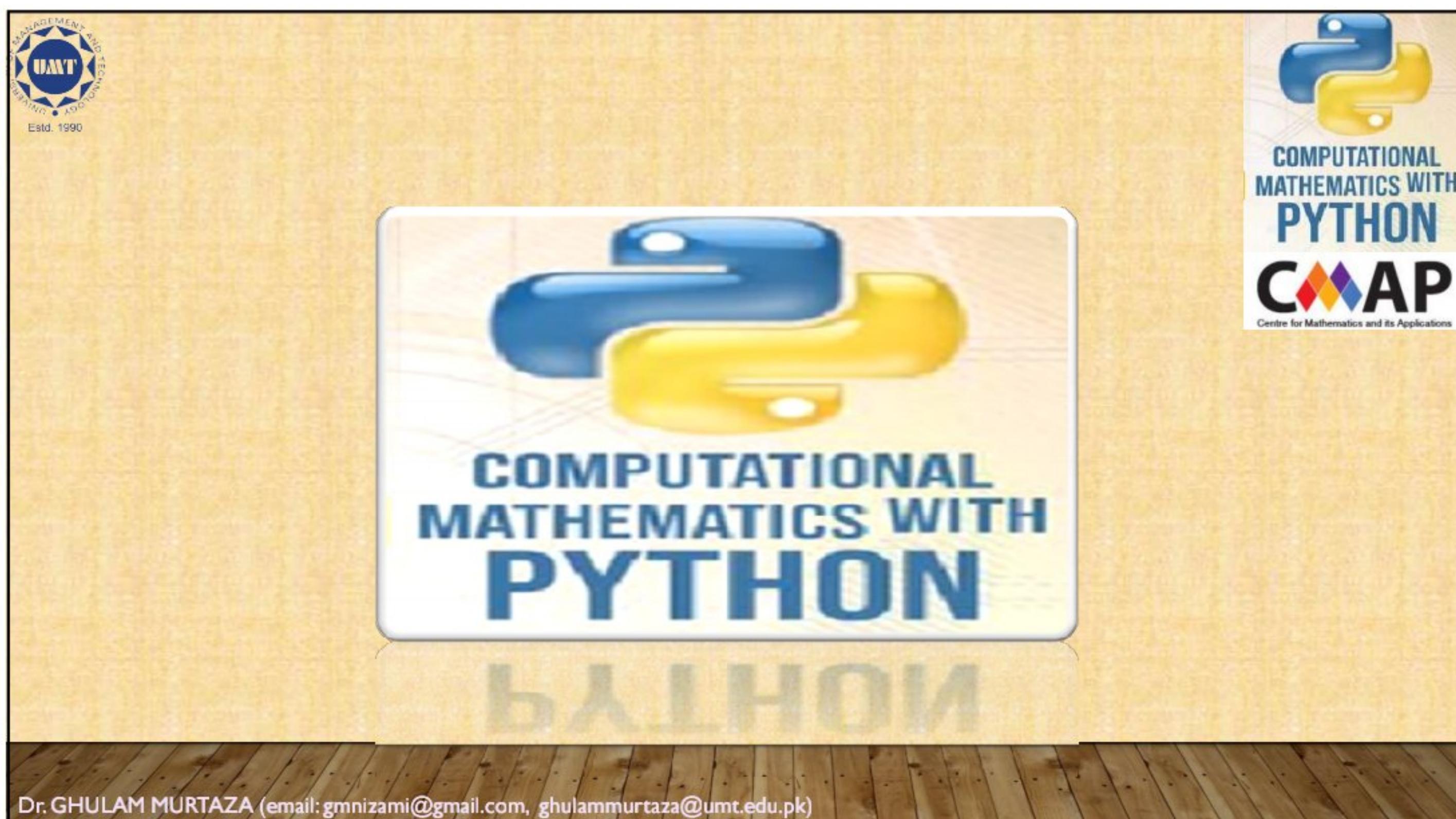


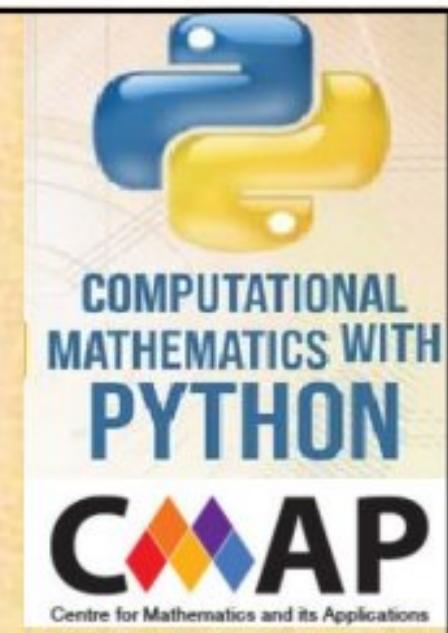
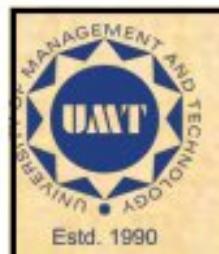
Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



FUNCTION	TO DO	EXAMPLE	 COMPUTATIONAL MATHEMATICS WITH PYTHON <b>CMap</b> Centre for Mathematics and its Applications
add()	Adds an element to the set	myset.add(element)	
clear()	Removes all the elements from the set	myset.clear()	
copy()	Returns a copy of the set	myset.copy()	
difference()	Returns a set containing the difference between two or more sets	myset.difference(otherset)	
Difference_update()	Removes the items in this set that are also included in another, specified set	myset.difference_update(otherset)	
discard()	Remove the specified item	myset.discard(element)	
intersection()	Returns a set, that is the intersection of two or more sets	myset.intersection(otherset)	
intersection_update()	Removes the items in this set that are not present in other, specified set(s)	myset.intersection_update(otherset)	
isdisjoint()	Returns whether two sets have a intersection or not	myset.isdisjoint(otherset)	
issubset()	Returns whether another set contains this set or not	myset.issubset(otherset)	
issuperset()	Returns whether this set contains another set or not	myset.issuperset(otherset)	
pop()	Removes an element from the set	myset.pop()	
remove()	Removes the specified element	myset.remove(element)	
symmetric_difference()	Returns a set with the symmetric differences of two sets	myset.symmetric_difference(otherset)	
symmetric_difference_update()	inserts the symmetric differences from this set and another	myset.symmetric_difference_update(otherset)	
union()	Return a set containing the union of sets	myset.union(otherset)	

Dr. GHULAM MURTAZA



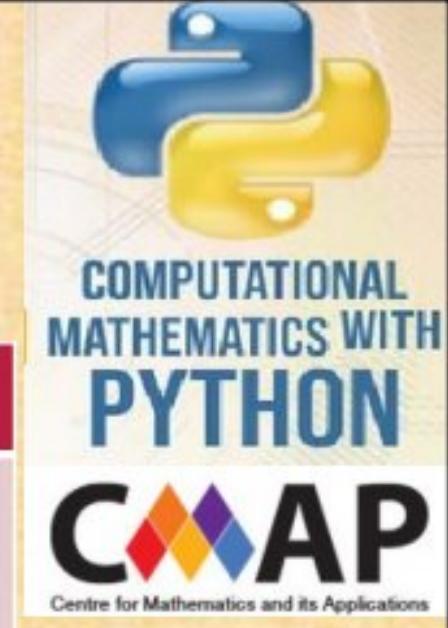
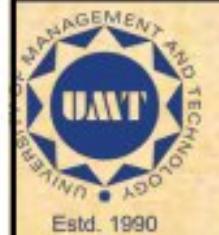


## TUPLE

A tuple is a collection of data (may be of different datatype) which is immutable.

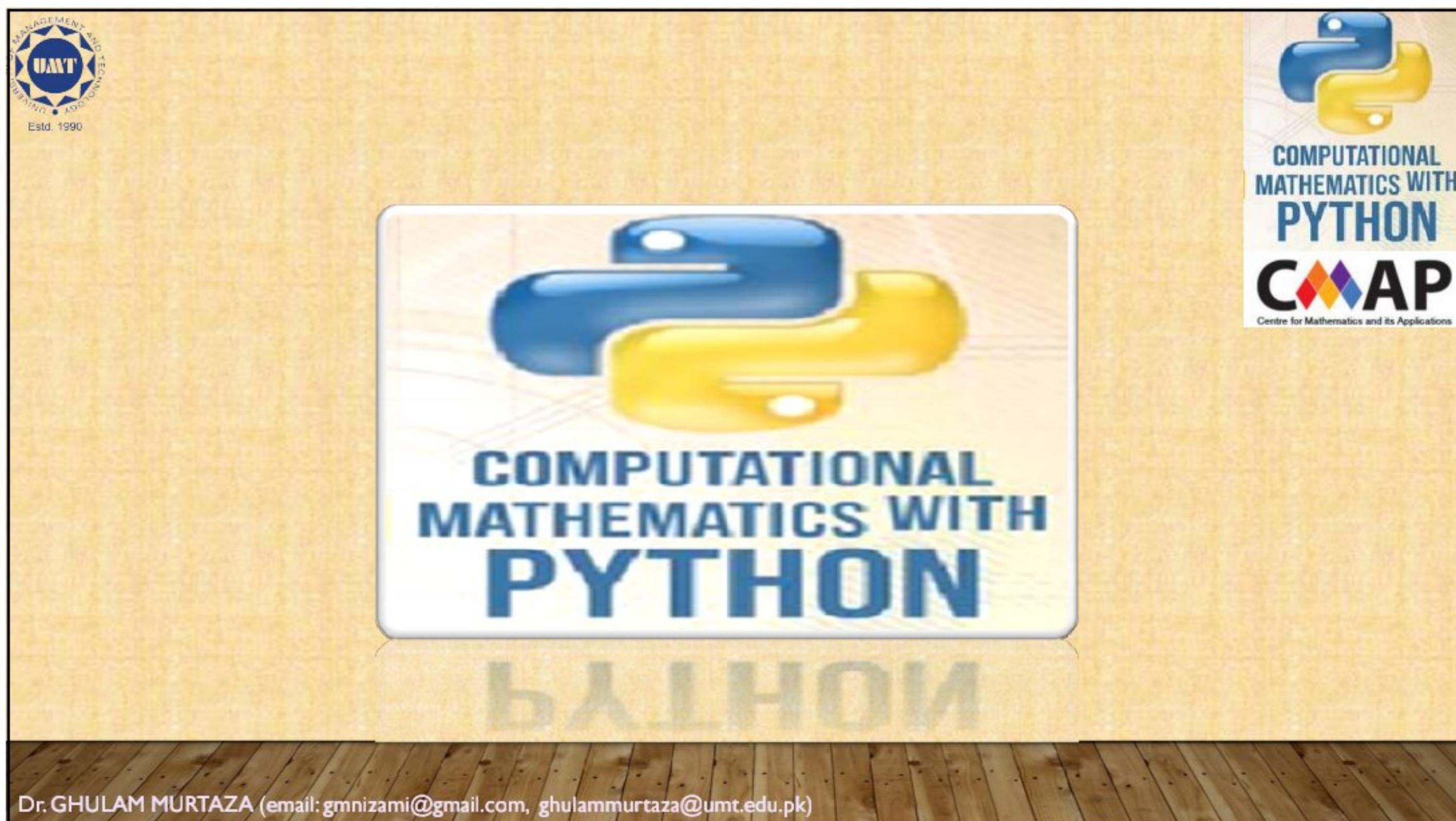
A tuple is in many ways similar to a list; one of the most important differences is that tuples can be used as keys in dictionaries and as elements of sets, while lists cannot.

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



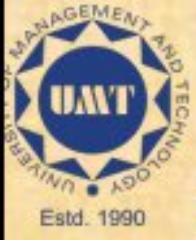
Function	To do	Example
len()	Returns number of items in the tuple	len(mytuple)
index()	Returns the index of the value specified	mytuple.index(item)
count()	Returns the number of occurrence of the specified item in the tuple	mytuple.count(item)

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)



Type	Access	Order	Duplicate values	Mutability
List	index	Yes	Yes	Yes
Dictionary	key	Yes	Yes	Yes
Set	No	No	No	Yes
Tuple	Index	Yes	Yes	No

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)

 **FUNCTIONS**

A function is a block of code that performs specific task. It executes only when it is called. Functions are classified into the following types:

- Built-in Functions

Functions that are readily available within the python library are known as built-in functions.

There are many built-in functions available. `print()`, `input()`, `list()`, `dict()` etc., are some of the built-in functions in python.(for more details [see this](#))

- User-defined Functions

Functions that are created and defined by the user are known as user defined functions.

The `def` keyword is used to create a user defined function. One must call that function in order to use it. Function must be defined first before we call it. Otherwise it shows an error.

Dr. GHULAM MURTAZA (email: [gmnizami@gmail.com](mailto:gmnizami@gmail.com), [ghulammurtaza@umt.edu.pk](mailto:ghulammurtaza@umt.edu.pk))

  
**COMPUTATIONAL MATHEMATICS WITH PYTHON**  
**CMAP**  
Centre for Mathematics and its Applications



## Module vs Function

The main difference between a module and a function is that a module is a collection of functions that are imported in multiple programs and can do various tasks. A function is a small block of code and separates itself from the entire code and have a fixed functionality. This function can be used anywhere within the same program whereas modules can be used in multiple programs.

**Examples of modules:** [math](#), [cmath](#)

Dr. GHULAM MURTAZA (email: [gmnizami@gmail.com](mailto:gmnizami@gmail.com), [ghulammurtaza@umt.edu.pk](mailto:ghulammurtaza@umt.edu.pk))

  
**COMPUTATIONAL MATHEMATICS WITH PYTHON**  
**CMAP**  
Centre for Mathematics and its Applications

DEFINING FUNCTIONS

```
name           parameters
def functionName(param1, param2, ...):
    statement1
    statement2
    ...
    }
```

body

Dr. GHULAM MURTAZA (email: gmnizami@gmail.com, ghulammurtaza@umt.edu.pk)