Introduction to Python:

Lab#01

Instructor: Dr. Tassadaq Hussain

Research Assistant: Engr. Hadiqa Bashir

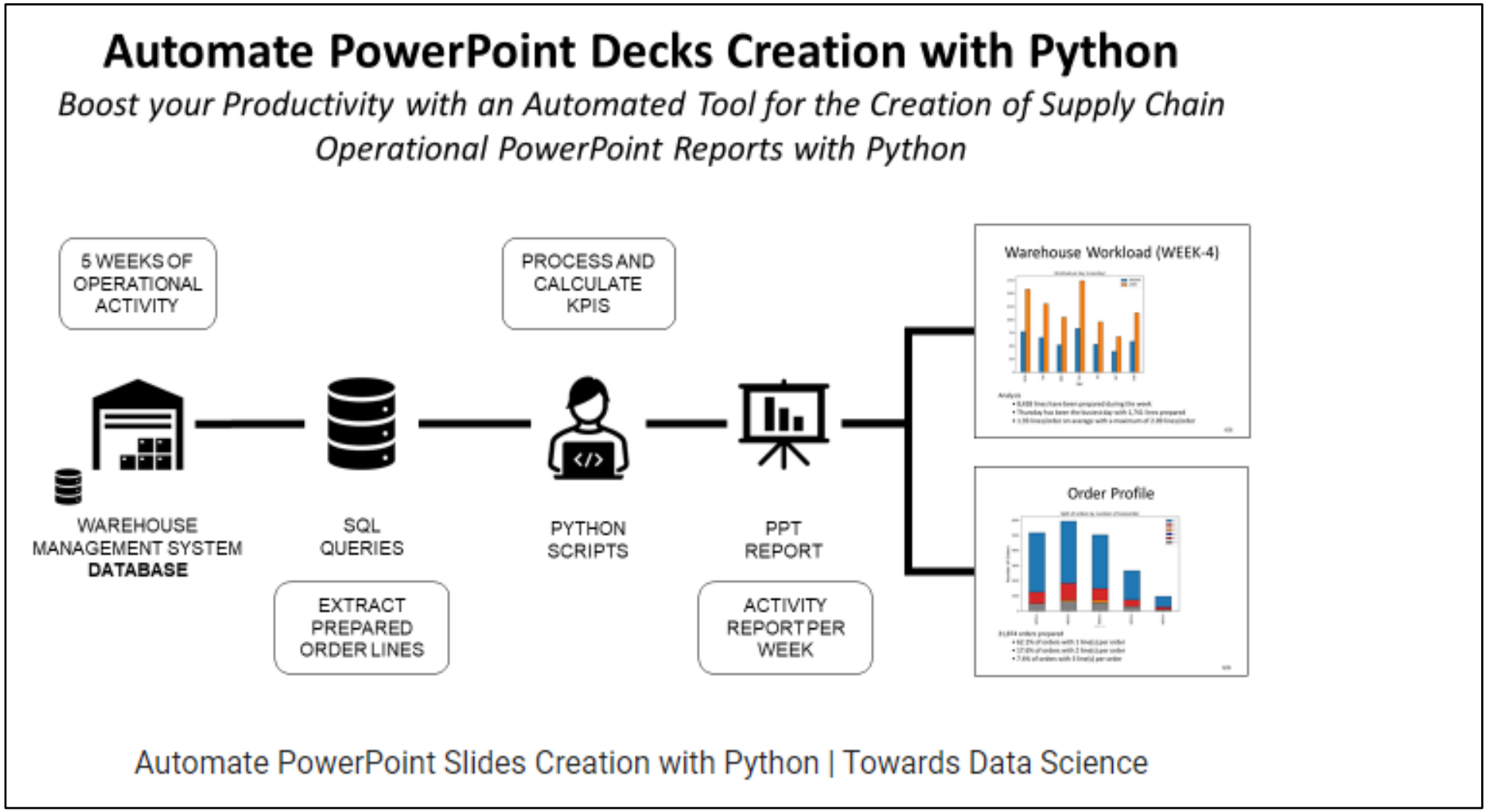
Email:hadiqa.bashir@namal.edu.pk

Contents:

Scope:

- Web applications
- Science
- Feature extraction
- Testing scripts
- Other (Emotion recognition, automatic Powerpoint reports)

Automate Powerpoint reports creation:



Python VS other languages:

- **Python**: Python code is read and executed by the Python interpreter one line at a time. It's not turned into machine code upfront, so it's generally slower for running programs directly.
- **Compiled Languages (e.g. C, C++, Java)**: These languages are transformed into machine code before running, which makes them faster when it comes to executing programs.

Basic Understanding of Data and Types

- Data Type:
- A data type represents the type or kind of data that a variable or object can hold in a programming language.
- It defines what operations can be performed on the data, how much memory it occupies, and how the data is stored.
- Examples of common data types include integers, floating-point numbers, strings, and booleans.

- Data Set:
- A data set is a collection of data points or records that are related in some way.
- It often represents a sample or a population of data used for analysis, research, or study.
- Data sets can be organized in various structures like tables, spreadsheets, databases, or even files.

- Data Structure:
- A data structure is a way of organizing and storing data in a computer's memory or storage devices.
- It defines how data elements are arranged, accessed, and manipulated.
- Data structures can be simple, like arrays and linked lists, or complex, like trees and graphs.

Introduction to Datatypes:

Python offers variety of datatype that includes int,float,string,bool,list,tuple,dictionary,set,complex,Nonetype,range.

Int(Integer): Integer datatype is use to express integer that include whole numbers, positive numbers, negative numbers.

Example:

```
[1] x = 34    # positive number
```

```
[2] x=-3      # negative number
```

Cont…

- **Float(floating point):**Floating point in python is used to represent numbers with decimal point.

  Example:

```
h= 0.44
f= 4.33
a=h+f

g= 6.77
```

- **str(string):** Strings are used to represent sequence of characters:

  Example:

```
string=" example"
length=len(string)
```

Cont…

- **Lists**:Python offers a range of compound data types often referred to as sequences. Lists represent ordered collection of data.
- **Example:**

```
mylist=[]               # empty list
mylist=[1,2,3]          # list of integers
mylist=[1,"text",8.1]   # multilist
```

- **Tuple**:A tuple in Python is similar to a list.But we cannot change the elements of a tuple onceit is assigned whereas we can change the elementsof a list.
  Example:

```
mytuple=()             # empty list
mytuple=(1,2,3)         # integers
mytuple=(1,"text",8.1)   # multituple
```

Cont…

- **Bool(Boolean):**Bool function is use to represent values either true or false.

  Example:

```
x = 10
y = 10
equal = (x == y)  # True
```

- **Nonetype:** This type is used to represent absence of value.

  Example:

```
value= None
```

Type Conversion:

- Python provides a special function called *type*, which allows us to figure out the *datatype* of any value .

Example:

```
type(1.0)
```

```
float
```

```
type("Hello")
```

```
str
```

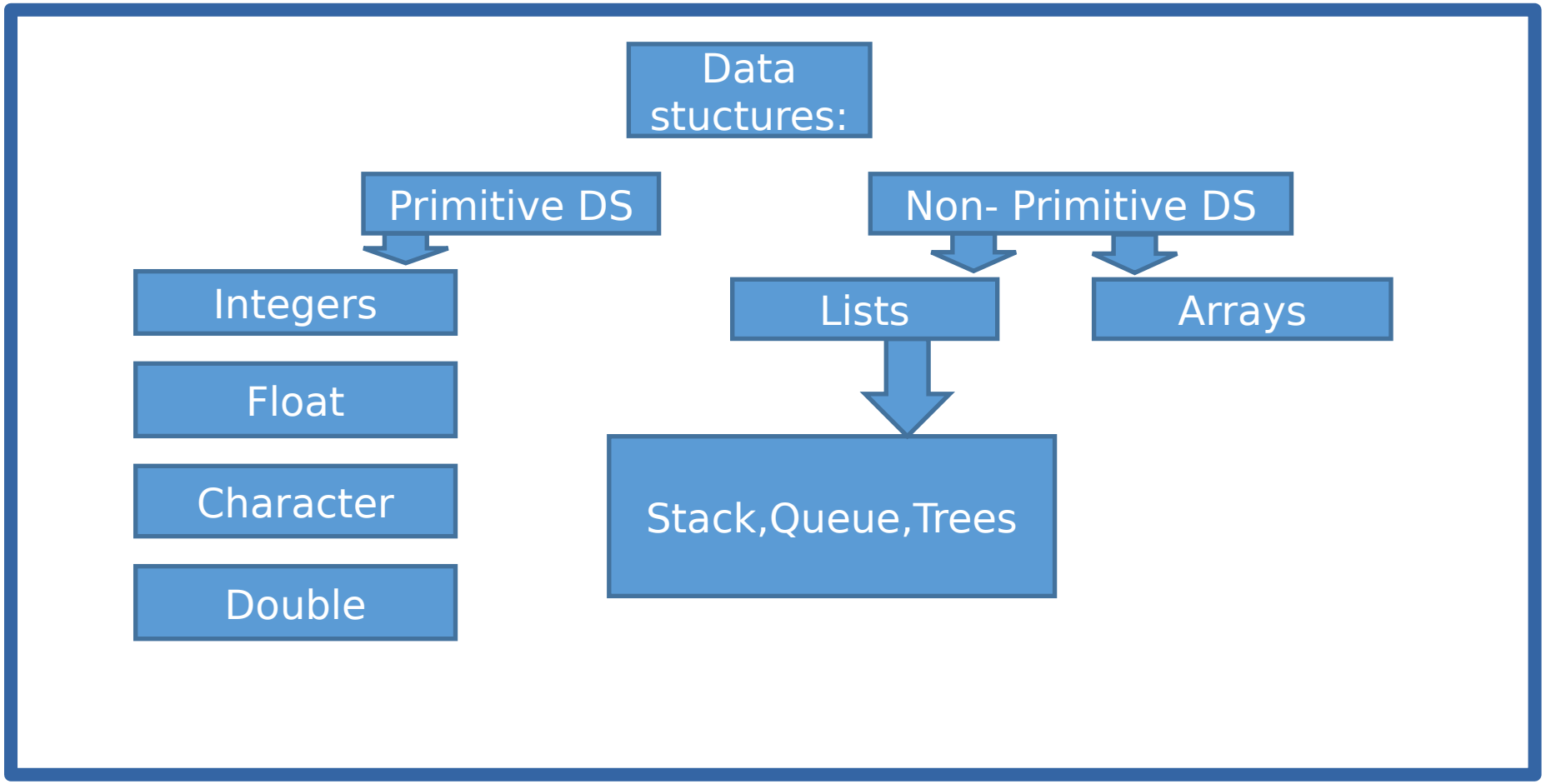
Where and when to use datatype:

- In any-programming datatypes are used to represent the type of data.
- Data types are used for type checking and validation to ensure that the right kind of data is used in specific contexts.
- Data type are also used to check memory allocation.
- To perform arithmetic operations, it is important to know the type of data.

Data structures:

- Data structures are used to store data in an organize form.
- They are fundamental for managing, storing, and retrieving data in a way that meets specific requirements and optimizes various operations like insertion, deletion, and search.

Classification of Data-structures:



Arrays:

- Arrays use single variable to store multiple values.
- Array can store multiple values but of the same data type.

Example:

- If we have list of items we can store in an arrays suppose to store marks of all subjects of a student.
- e.g: array_marks[5]={ 12, 45, 90,60,100}

Array Methods:

Python provide several methods that can be applied on lists/arrays:

insert().	**To insert any element at end of list.**
append()	Add any element to the end of list.
clear()	Removes all elements from list.
copy()	Returns copy of the list

Type Casting:

- **Type casting:**Type casting include how we can change one data type to another for example python int() function take value in float and return it as a int type objects.

  Use: To convert from one datatype to another.

  Examples:

```
number = 9  #int
float_number = float(number)
```

```
x = "1.2"
y = float(x)
print(y)
```

```
1.2
```

Data management

- Data management involves handling data efficiently.
- It involves various steps:

1. Choose appropriate datastructures.
2. CSV file handling.
3. Cleaning and pre-processing data.
4. Error handling.

Note: As a reefrence I will show my own fyp code as it is directly related here.

How to read files?

a)**Reading CSV, Txt Files etc.**

- We have to create csv file and upload it on drive,then using python command access it in your code.

Example:(We will discuss other examples in the lab).

```
from google.colab import files
uploaded = files.upload()
```

```
import pandas as pd

df = pd.read_csv('data.csv')
```

Cont…

b) Data Storage (SQL).First import the library and then connect like in the given example below:

Example:

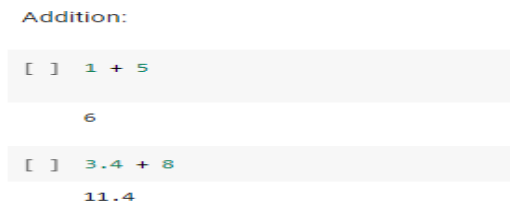
```
import mysql.connector

conn = mysql.connector.connect(

    password='your_password',
    database='your_database'
)
```

c)Reading from external port (USB, Serial): Explore on your own and let’s discuss in the class.

Arithmetic operators and expressions(along with some practice):

- Addition: to add two variable:

Example:



- Subtraction: To subtract two variable:

Example:

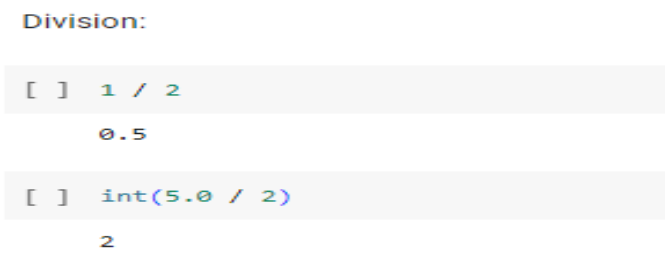
Cont…

- Multiplication: To multiply two numbers.

Example:

- Division: To divide two numbers.

Example:



Function and its’s working:

- Defination:
- Function are the basic building block of programming language. Function provide reusability. Function in python are similar to functions in other languages except the syntax.

Advantages of functions:

- **Abstraction:**

Function provide abstraction .

- **To reduce complexity:**

Function can break a complex program into smaller parts that are easy to understand.

- **Re-usability:**

No need to define function again and again,once we defined we can use it the program .we just need to call.

Understanding function.

- Defining function
- Function parameters
- Calling function
- Function body
- Returning values

Cont…

- In python , most of the functions are ‘built-in’ and others are the ones that we ‘ include’ using libraries.
- The use of the brackets type () for function calls.

```
2 ** 6
```

```
64
```

```
pow(2, 3)
```

```
8
```

Creating own function (example):

Write a function to find smallest number in the array(basic).

- Defining Function.

```
def smallestnum(array): # Define  function
```

Iterating through loop and return variables:

```
if len(array) == 0:
    return None

smallest = array[0]

for element in arr:
    if element < smallest:
        smallest = element

return smallest
```

Function call:

```
# Call
myarray = [1, 3, 6, 7, 8, 4, 10]
output = find_smallest(my_array)
```

Print Result:

```
# Print the result
print("smallest number is:", output)
```

Importing different Libraries:

- NumPy: for numerical and array operations.
- Math- provides mathematical functions and constants.
- Pandas-for data manipulation and analysis.
- Sklearn- Include tools for classification ,clustering etc.
- Other libraries are also available(Explore on your own for better use and understanding).

Numpy:For numerical and array operations.
(example):

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5,6,7,8])
mean = np.mean(arr)
total = np.sum(arr)
print("Mean:", mean)
print("Sum:", total)
```

Pandas(example):

- We can access different data from csv file using pandas.

```
import pandas as pd
df = pd.read_csv('data.csv')
df.dropna()
```

How to use module/libraries?

- The interpreter will considered all text as comment after #.
- Some basic examples(regarding libraries and modules).

```
import numpy  # Import the standard module numpy
```

Print & visualization:

- Print():In python print() is used to print the results:

Examples:

```
h = 0.44
f = 4.33
a = h + f
print("The value of a is:", a)
```

```
print("Hi!")
```

```
Hi!
```

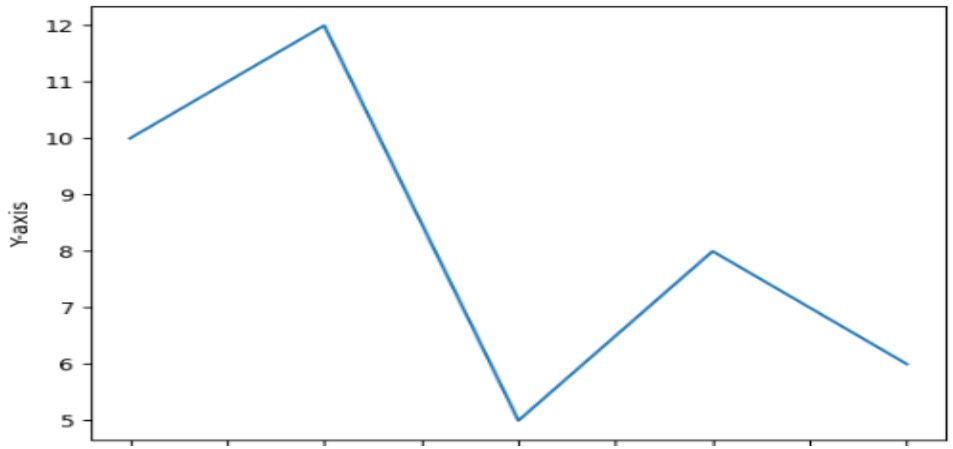
Cont…

- Visualization:To see the plotted graph or visualize other things we use matplot library.

Example:

```
y = [10, 12, 5, 8, 6]
plt.plot( y)
plt.ylabel("Y-axis")
```

```
Text(0, 0.5, 'Y-axis')
```



Practice problems:

- Simple Calculator using python .
- Fibonacci sequence using python functions.
- To create our own library.(Can be graded).

References:

- https:sujithkumar9212301/introduction-to-python-36647807
- https://www.udemy.com/course/learn-python-with-google-colab-a-step-to-machine-learning/learn/lecture/17757444#overview
- http://wwwd)ython.org/doc/
- Programming in python(basics)
- https://realpython.com/python-numbers/
- https://www.programiz.com/python-programming/array