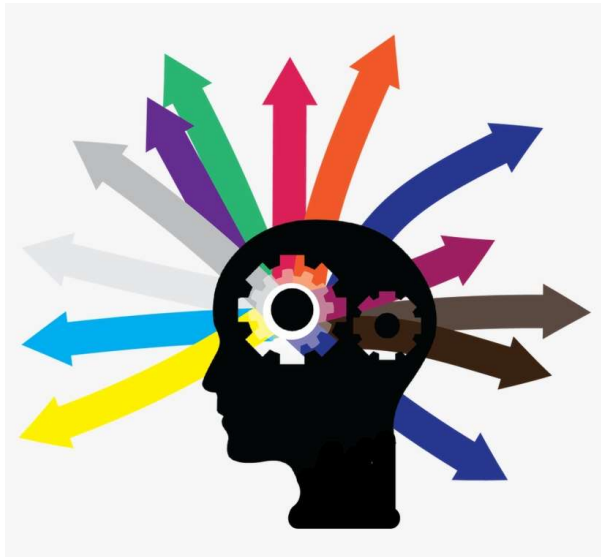# INTRODUCTION of Python

Lecture 1: Introduction

- **M.Sc. Riyadh Seed Agid**
- **Salahaddin University – Erbil**
- **riyadh.agid@su.edu.krd**

Computer Programming Course **Learning Outcomes**

1. To be able to understand fundamental of python programing.

2. To be able to creating code to solve physics problem.

3. To be able to plotting

Computer Programming Course **Outline**

Lecture 1	Introduction	Lecture 2	Fundamental
Computer, Programming, Python, Installation of Python , Interpreter, and Compiler.		Basic Math, Addition, Subtraction, Multiplication, Division And Exponentiation, Operators, Comparison, Advanced Mathematical.	
Lecture 3	Numpy package	Lecture 4	Arrays in python
Library, Numeric python, Scientific python, Matplotlib, Variables.		Arrays, Append, Extend, Arrange, Linspace, Operations With Array, Vector Algebra, Dot And Cross Product Of Vectors.	
Lecture 5	Matplotlib package	Lecture 6	Style plotting
Basic plotting, Markers, Line plots, Colour, Marker Size, Line Width, Labels and Title.		Formatting, style, grid,, legend, Subplots, Scatter Plots, Alpha.	
Lecture 7	Polynomial plotting	Lecture 8	Writing and plotting Python straight line program
Array statistics, roots of polynomial, Histograms, Bars, Pie Charts, Python strings.		Straight line , Input and data plotting, Saving a program.	
Lecture 9	Opening files in Python		
Reading and writing files, Deleting and renaming files in Python			

➤ A computer is a Machin that stores pieces of information.

➤ A computer also moves, arranges, and controls that information (or data).

➤ programming It is a set of orders that are written in English sequentially and logically in one of the programming languages.

➤ When these orders are executed, they are translated into machine language so that the computer can understand and execute them.

INTRODUCTION

What is Python?

❑ **Python** is a popular programming language, high level, objective oriented programming language (invented by Guido Van Rossum and released in 1991).

Guido van Ros...

❑Scripting means programming at a high and flexible abstraction level using source code (i.e. scripts) that are interpreted rather than compiled.

❑Python supports modules and packages, which encourages program modularity and code reuse.

Why python

❑ Python is an easy to learn freely available, high level.

❑ Python is an interactive language. This means we don't need to compile and then run the code but execute each line as you type it.

❑ Python is open source and free software runs on Windows, Mac and Linux. This means that you will be able to download and install Python on any of PC or laptop.

❑ Python has syntax that allows developers to write programs with fewer lines than some other programming languages.

❑ Python has a built-in GUI (graphical user interface) library via Tkinter, which lets the programmer make simple graphical interfaces with little effort.

❑ Portability. Because python is for free, your code can run everywhere.

❑ Powerful. Because it is well designed , it is easier to transfer your idea into cod.

What can Python do?

The most common Python roles currently seem to fall into a few broad categories. Below are some of Python's most common applications today:

- ❖ developing websites and software

- ❖ GUIs (graphical user interface)

- ❖ Internet scripting

- ❖ Database programming

- ❖ Numeric and scientific programming

- ❖ Gaming, imaging, data mining, robots, …

➤ ForecastWatch.com

The software used to make the comparisons (between thousands of forecasters against actual climate to find their accuracy) is written in pure Python because it comes with standard libraries useful in collecting, parsing, and storing data from online sources.

➤ Frequentis

A software product that is used for air traffic control in many airports. This particular tool provides updates on the weather and runway conditions to air traffic controllers.

➤ Corel

PaintShop Pro is a product that many people have used over the years to grab screenshots, modify their pictures, draw new images, and perform a lot of other graphics-oriented tasks.

➤ Honeywell

Documenting large systems is expensive and error prone. Honeywell uses Python to perform automated testing of applications, but it also uses Python to control a cooperative environment between applications used to generate documentation for the applications

➤ United space Allianc

This company provides major support to NASA (National Aeronautics and Space Administration) for various projects, such as the space shuttle. Python was chosen over languages such as Java and C++ because it provides dynamic typing and pseudo-code–like syntax and it has an interpreter.

Installation of Python

www.python.org

Step 1: Select (2.7.18) Version of Python to Install

Step 2: Download Windows x86-64 MSI installer

Step 3: Run Windows x86-64 MSI Installer

Installation of Python

- Adding Python to the "**System Path Variable**", it allows Python to run scripts without any problem. To do that, you need to follow these steps:

  - Open "**Control Panel**"
  - Select "**System**"
  - Select "**Advanced System Setting**"
  - Select "**Advanced**" tab
  - Then select "**Environmental Variables…**"

- Once you have the "**Environment Variables…**" window open, search for "**path**" and click on it.

- Type the following to the end of the path directory:
                    ;C:\Python27

- Then press "**OK**" to close window and to apply the changes.

Getting Packages

❑Open "**Command Prompt**"

❑Type the following command:

cd C:\Python27\Scripts

❑Then type the following command to install "**matplotlib**" package:

pip install matplotlib

❑Repeat the last command to install "**numpy**" and "**scipy**" packages, separately.

**pip install numpy**

pip install scipy

An interpreter reads a high-level program and executes it, meaning that it does what the program says. It processes the program a little at a time, alternately reading lines and performing computations

SOURCE CODE → INTERPRETER → OUTPUT

A compiler reads the program and translates it completely before the program starts running.

SOURCE CODE → COMPILER → OBJECT CODE → EXCUTOR → OUTPUT
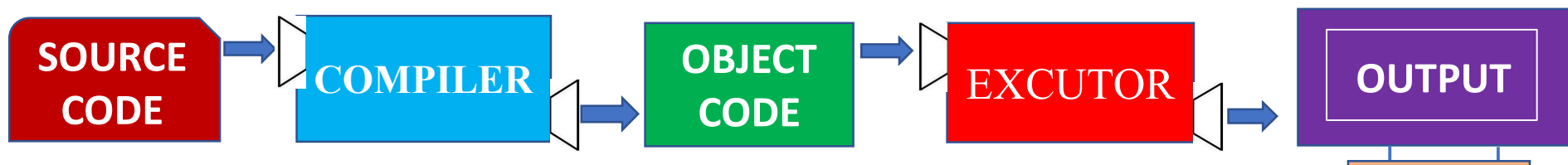
Interpreter	VS	Compiler

- ❖ Translates program one statement at a time.
- ❖ It takes less amount of time to analyze the source code but the overall execution time is slower.
- ❖ Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.
- ❖ Programming language like Python, Ruby use interpreters.

- ❖ Scans the entire program and translates it as a whole into machine code.
- ❖ It takes large amount of time to analyze the source code but the overall execution time is comparatively faster.
- ❖ It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.
- ❖ Programming language like C, C++ use compilers.

Thank you