

Summer 2023 Training Course:
Artificial Intelligence Applications in Structural Engineering
Week 1: **Python Functions/Classes - 03**

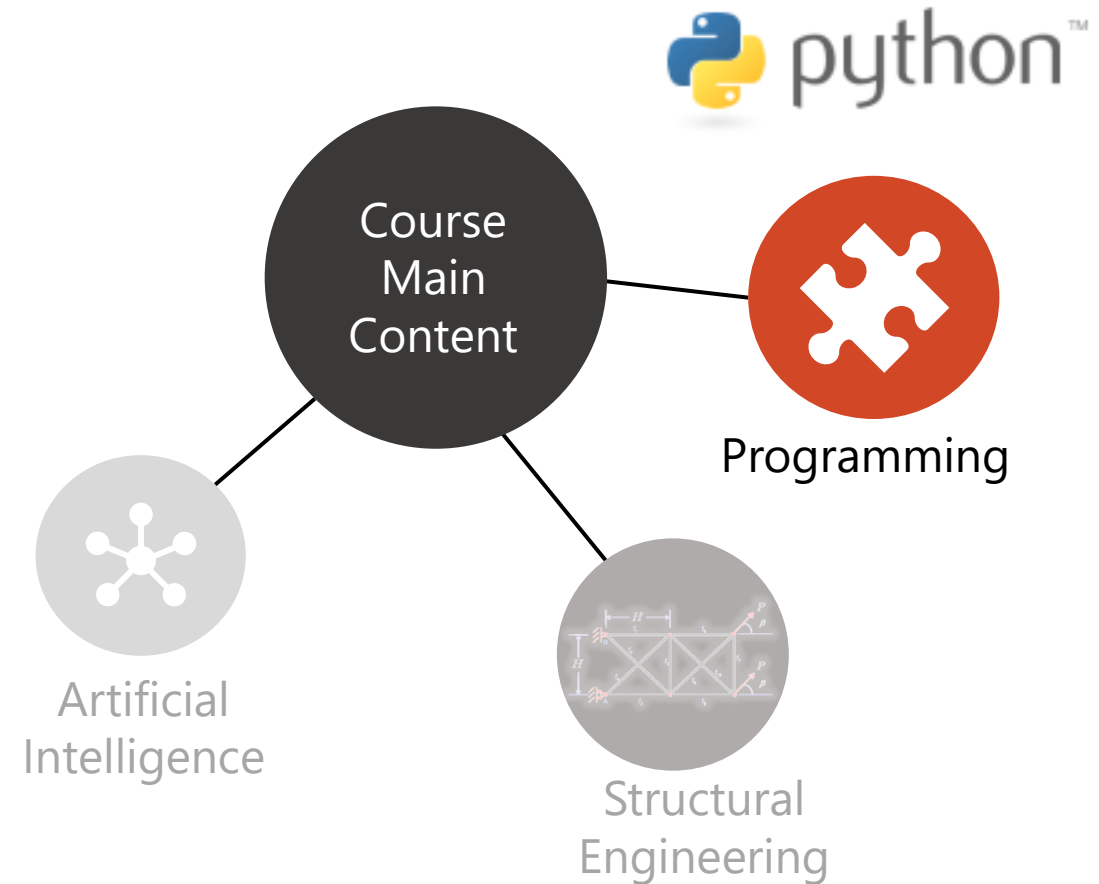
Dr. Ahmed A. Torky – Civil Engineering Department



Week 1 – Day 3

Programming with Python:

1. Github Linking
2. Python Functions
3. Python Classes
4. Best Practices
 - Type-hinting
 - Doc-strings



GitHub – Connection

This is a simplified way to attain files and folders from the AI Applications in Structural Engineering **Repository on Github**.

I recommend checking this video in the future on your own: <https://youtu.be/9bJfG1C2Z3M>

****DO THIS PART ONLY ONCE:****

1. Make an account on Github FIRST!
2. Fork the BUE-AI-Structural-Engineering-Beginner repo in your browser.
3. Clone a Copy of YOUR Fork to your local computer. In PowerShell (Windows) or in terminal (MacOS), cd to a directory you want to place the repo and then type:
 - `git clone https://github.com/YOUR_NAME/BUE-AI-Structural-Engineering-Beginner.git`

NOTE: Make sure you change the part "YOUR_NAME" in the command below!

Continued in the Jupyter Notebook file...

Python – Functions

A function is a reusable block of code that performs a specific task. Functions help to organize code and make it more modular, readable, and maintainable. A function can take input arguments, perform some operations on them, and return a result.

Function to get moment of inertia

```
1 def moment_of_inertia(b, h):  
2     """Calculate the moment of inertia of a rectangular cross-section"""  
3     I = (b * h**3) / 12  
4     return I
```

Calling the function

```
1 # Calculate the moment of inertia of a rectangular cross-section with width 10 cm and height 20 cm  
2 I = moment_of_inertia(10, 20)  
3 print("The moment of inertia is:", I, "cm^4")
```

Python – Classes

A class is a blueprint for creating objects. A class defines a set of attributes and methods that describe the behavior and properties of the objects created from the class. In other words, a class is a template for creating objects with specific characteristics and behaviors.

Class to create objects that compute I_x

```
1 class RectangularCrossSection:
2     """A class that represents a rectangular cross-section"""
3     def __init__(self, width, height):
4         """Initialize a new RectangularCrossSection object with the given width and height"""
5         self.width = width
6         self.height = height
7
8     def moment_of_inertia(self):
9         """Calculate the moment of inertia of the rectangular cross-section"""
10        I = (self.width * self.height**3) / 12
11        return I
```

Calling the Class to get an instance of the Object

```
1 # Create a new RectangularCrossSection object with width 10 cm and height 20 cm
2 section = RectangularCrossSection(10, 20)
3
4 # Calculate the moment of inertia of the rectangular cross-section
5 I = section.moment_of_inertia()
6 print("The moment of inertia is:", I, "cm^4")
```

Python – Best Practices

Here are some best practices in Python coding to keep in mind:

1. **Use descriptive variable names:** Choose variable names that clearly describe their purpose and avoid using single-letter variable names.
2. **Write docstrings:** Use docstrings to document the purpose, inputs, and outputs of functions and classes.
3. **Follow PEP 8 style guide:** Adhere to the PEP 8 style guide for Python code. This includes guidelines for indentation, line length, naming conventions, and more.
4. **Use comments sparingly:** Use comments to explain complex code or to provide context, but avoid over-commenting code that is already self-explanatory.
5. **Avoid global variables:** Use local variables within functions and avoid using global variables unless absolutely necessary.
6. **Use list comprehensions:** Use list comprehensions instead of for loops to create new lists or filter elements from existing lists.
7. **Use exception handling:** Use exception handling to gracefully handle errors and prevent crashes.

Try it out – Exercise!

OPEN THE NOTEBOOK:

03-Python Functions_Classes.ipynb

Try it out – Assignment!

1. Complete the Notebook Exercises.
2. Test these Exercises in Spyder.