Course content:

Day 1- Variables, data types and operators. Conditional statements.

1. Quick intro: What is Python?
2. Setting up the environment.
3. IDLE: How to use; editor and interactive modes.
4. Variables
    - What are variables?
    - Data types. int, float, string, bool. The "None" type.
    - Variable naming rules.
    - Type conversion.
5. Operators
    - Arithmetic and logical operators.
    - Boolean logic.
    - Operator precedence.
6. Basic input and output (I/O)
    - The "print" and "input" functions.
    - The "format" function.
7. Comments.
8. Conditionals
    - if statements.
    - if… else statements.
    - if… elif… else statements.
9. Modules and libraries
    - Difference between modules, packages libraries and frameworks.
    - The Standard Library.
    - PIP.
    - Installing the matplotlib library. Basic graph plotting.
    - "import this". The Zen of Python.

Day 2- Loops and collections. Lists, tuples, sets, dictionaries.

1. Loops
    - while loops.
    - for loops.
    - The "range" function.
    - break and continue.
    - for… else loops.
    - Infinite loops.
    - The "pass" statement.
2. Lists
    - Creation and usage.
    - Indexing. - Appending, inserting, removing and modifying list items.
    - The "len" function. - "in" and "not in". The "index" method.
    - List slices.
    - Strings as lists.
    - Looping through lists.
    - List comprehensions.
3. Tuples
    - Tuples as immutable data types
    - Tuple packing and unpacking
4. Dictionaries and sets
    - Dictionaries. Keys and values.
    - Adding and modifying dictionary items.
    - Finding keys. "in" and "not in".
    - Looping through dictionaries.

- Sets. Set functions.
- Operations on sets. Union, intersection, difference and symmetric difference.
- Differences between dictionaries and sets.

Day 3- Introduction to functions. Functional programming.

1. Functions
    - Why functions? Need for functions.
    - Function basics. Parameters and arguments.
    - The “return” statement.
    - In-built functions.
2. Documentation. Docstrings.
3. More on functions
    - Functions as variables/objects.
    - Default argument values. “args” and “kwargs”.
    - Function overloading.
    - Variable scope.
    - Value and reference types.
4. Functional programming
    - Pure functions.
    - Lambdas. Introduction to lambda calculus.
    - “list”, “map” and “filter” functions.
    - Function decorators.
    - Generators.
    - The “itertools” module.

Day 4- GUI and Introduction to Object-oriented programming.

GUI

- TKINTERS

1. Why OOP? OOP design
    - Overview. - Real world examples.
    - Features in OOP.
2. Classes and objects
    - Difference between classes and objects.
    - Class creation.
    - Object instantiation.
    - Constructors. - “self”.
3. Properties and methods
    - States.
    - Behaviours.
    - Properties. The @property decorator.
    - @classmathod and @staticmethod.
4. Magic methods. Operator overloading.

Day 5- Hands - on