1. Module 1: Basic Concepts

- Basic concepts
- Interpreter, compiler, lexis, semantics, syntax, Keywords, etc.
- Literal - Integer, Boolean, Floats, scientific notation, string
- comment
- print()
- input()
- Other numeral systems - binary, decimal, octal, and hexadecimal
- Aritmetic(numeric) operators -> +, -, *, /, //, %, **
- String Operators: +, *
- assignments and shortcut operators

2. Module 2: Data Types, Evaluations, and Basic I/O Operations

- operators: unary and binary, priorities and binding
- Boolean operators: or, and, not
- Boolean expressions
- relational operators (== != > >= < <=), building complex Boolean expressions
- accuracy of floating-point numbers
- basic i/o (input and output) operations using - input(), print(), int(), float(), str(), len() functions
- **format print**() output with **end=** and **sep=** arguments
- type casting
- basic calculations
- simple strings: constructing(create), assigning, indexing, immutability

3. Module 3: Control Flow – loops and conditional blocks

- conditional statements: if, if-else, if-elif, if-elif-else
- multiple conditional statements - multiple if statement
- the pass instruction
- Loops: while, for, range(), in
- **Iteration** - iterating through sequences
- loops continued: while-else, for-else
- nesting loops and conditional statements
- controlling loop execution: **break**, **continue**

4. Module 4: Data Collections – Lists, Tuples, and Dictionaries

- simple **lists**: constructing vectors, indexing, and slicing, the len() function
- lists in detail: indexing, slicing, basic methods (append(), insert(), index()) and functions (len(), sorted(), etc.), del instruction, iterating lists with the for loop, initializing, in and not in operators, list comprehension, the **difference between copying and cloning**
- lists in lists: matrices and cubes
- **tuples**: indexing, slicing, building, immutability
- **tuples vs. lists**: similarities and differences, lists inside tuples and tuples inside lists
- dictionaries: building, indexing, adding and removing keys, iterating through dictionaries as well as their keys and values, checking key existence, keys(), items(), and values() methods
- strings in detail: escaping using the \ character, quotes, and apostrophes inside strings, multi-line strings, basic string functions.

5. Module 5: Functions

- define and call(invoking) your own functions(user-defined functions) and generators
- return and yield keywords, returning results,
- the None keyword,
- **recursion**
- parameters vs. arguments,
- positional keyword and mixed argument passing,
- default parameter values
- converting generator objects into lists using the list() function
- **name scopes**, name hiding (shadowing), the **global** keyword

```
for all_days in range(start, end):
    print("Thank you for helping me!")
```

```
if do_practice:
    print("I happy!")
else:
    print("I sad!")
```