PYTHON PROGRAMMING

---

1. Recall and describe the basic concepts of programming algorithms and flowcharts to solve problems:
- Define algorithms and flowcharts
- Identify the steps involved in solving a problem using algorithms and flowcharts
- Create and interpret flowcharts for problem-solving

2. Apply Python operators, expressions, and statements to perform basic operations and calculations:
- Identify and use arithmetic, comparison, and logical operators in Python
- Construct and evaluate expressions using operators
- Write and execute Python statements to perform tasks

3. Utilize sequence data types (set, tuple, list, dictionary, and arrays) in Python for data manipulation and organization:
- Differentiate between set, tuple, list, dictionary, and arrays in Python
- Create, access, update, and manipulate data in different sequence data types
- Apply appropriate sequence data type for specific data storage needs

4. Implement functions and parameterized functions to modularize and organize code in Python:
- Define functions with and without parameters in Python
- Call and use functions to perform specific tasks
- Compose and reuse functions for code modularity and organization

5. Perform file processing operations in Python to read and write data from/to files:
- Open, read, write, and close files in Python
- Handle file processing errors and exceptions
- Implement file processing operations for data storage and retrieval

6. Manage scope and utilize modules in Python for code structuring and reusability:
- Explain the concept of scope in Python programming
- Create and use modules to structure and organize code
- Import and utilize modules for code reusability

7. Apply basic libraries in Python for advanced functionality and data manipulation:
- Identify and import basic libraries in Python (e.g., NumPy, Pandas, Matplotlib)
- Utilize basic libraries for mathematical operations, data analysis, and visualization
- Apply basic libraries to enhance functionality and efficiency in Python programming.