- ☐ I can explain what a code generator is and why they're useful
- ☐ I can define an algorithm
- ☐ I can write pseudocode to write out the logic of an algorithm
- ☐ I can explain the characteristics of and use the following programming fundamentals to create simple programs:
    - Data types
        - Text: string, character,
        - Numeric: *int, float*
        - *Boolean*
    - Data structures
        - One-dimensional arrays
        - Lists
        - Records (varying data types, field index)
- ☐ I know the characteristics of functional and non-functional requirements, constraints, and scope
- ☐ I can use the following design tools to represent an algorithm/program/piece of functionality
    - Mock-ups
    - Input-process-output (IPO) charts
    - Flowcharts/pseudocode
- ☐ I can describe key legal requirements relating to intellectual property and copyright, such as licensing, code plagiarism, and code theft
- ☐ I can explain and implement the following principles of Object-Oriented Programming:
    - Abstraction
    - Encapsulation
- ☐ I know and can use the key features of python, such as variables, accessing/storing data in variables, iteration, arithmetic and logic operators, functions, methods and procedures
- ☐ I can explain and use naming conventions, e.g. camel casing or Hungarian notation
- ☐ I understand and can implement documentation such as comments
- ☐ I know and can describe the following validation techniques for data:
    - Existence checking
    - Type checking
    - Range checking
- ☐ I can utilise debugging and testing techniques for checking software solutions function correctly, such as:
    - Test tables
    - Breakpoints
    - Debugging output statements