Topics/Content/Outline
Topics include: array lists, coding of graphical user interfaces, event handling, exception handling, sorting and search algorithms, binary files, enumerated types.

Outcomes
Outcome 1: Identify the fundamental data requirements of an intermediate level program. • Select and use relevant data types. • Explain reasons for using different types of variable declarations (static, local and global). • Pass parameters into subprograms and return values from subprograms to ensure data independence. • Explain the difference between call-by-reference parameters and call-by-value parameters • Explain reasons for using constants and enumerated types. • Explain the reasons for exception handling.
Outcome 2: Apply the logic structures of the language. • Write programs that use complex nested decisions and loops. • Use text and binary files for input and output • Write code for graphical interfaces • Find program logic errors using the debugging features of an integrated development environment. • Fix errors in program logic.
Outcome 3: Select and use intermediate-level data structures and available algorithms. • Define and use intermediate-level data structures. • Describe the relative advantages and disadvantages of different data structures (for example, array storage versus linked lists) and different forms of file access for handling records. • Identify and use search and sort algorithms. • Compare efficiency of search and sort algorithms applied to intermediate level data structures.
Outcome 4: Write a complete program whilst adhering to available coding standards. • Develop a design to meet the requirements of the problem. • Write a program using the resources supported by the programming language. • Explain the reasons for selecting the resources. • Adhere to the principles of writing code to ensure reuse and maintenance. • Internally document code according to a given standard. • Create and use a test plan to test a program thoroughly.

Assessment:

Weighting	Nature of assessment	Learning Outcomes
30%	Programming assignment which includes design, coding and testing of an intermediate level application	1,2,3,4
30%	Practical Tests which require the writing and modification of intermediate level program code	1,2,3,4