

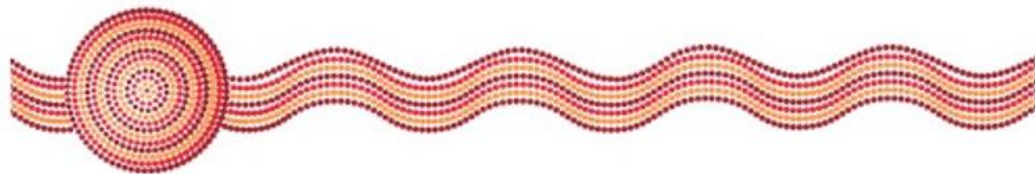
# VCE Applied Computing 2025–2028

Video 10

Background to Unit 3 Outcome 1  
Software Development

# Acknowledgement of Country

The VCAA respectfully acknowledges the Traditional Owners of Country throughout Victoria and pays respect to the ongoing living cultures of First Peoples.



# VCE Applied Computing 2025–2028

Video 10

Background to Unit 3 Outcome 1  
Software Development

# Purpose of this presentation

- Overview of Unit 3 Outcome 1 Software Development
- Major changes to Unit 3 Outcome 1
- Software tools
- Outcome statement
- Key knowledge
- Key skills
- Assessment task

# Unit 3 Outcome 1



VICTORIAN CURRICULUM  
AND ASSESSMENT AUTHORITY



# Changes to Unit 3 Outcome 1

- Emerging trends
- Object-oriented programming
- Errors
- Updated assessment task (SAC)

# Unit 3 Outcome 1

## Software tools

Students are required to use the following software tool:

- An appropriate object-oriented programming language.

# Unit 3 Outcome 1

## From the outcome statement

- Interpret teacher-provided solution requirements and designs and use appropriate features of an object-oriented programming language to develop working software modules.



# Unit 3 Outcome 1 – Key knowledge

- emerging trends in programming using artificial intelligence, including:
  - using prompts to generate code
  - automated debugging and testing of modules
  - code optimisation
  - responsible and ethical use of artificial intelligence tools
- characteristics of functional and non-functional requirements, constraints and scope

# Unit 3 Outcome 1 – Key knowledge

- design tools for representing modules, including:
  - data dictionaries
  - mock-ups
  - object descriptions
  - input-process-output (IPO) charts
  - pseudocode
- characteristics of data types, including:
  - text (character, string)
  - numeric (integer, floating point, date/time)
  - Boolean

# Unit 3 Outcome 1 – Key knowledge

- characteristics of data structures, including:
  - one-dimensional arrays
  - two-dimensional arrays
  - records (varying data types, field index)
- characteristics of data sources (plain text (TXT), delimited (CSV) and XML files), including:
  - structure
  - reasons for use

# Unit 3 Outcome 1 – Key knowledge

- principles of OOP, including:
  - abstraction
  - encapsulation
  - generalisation
  - inheritance

# Unit 3 Outcome 1 – Key knowledge

- features of a programming language, including:
  - local and global variables, and constants
  - data types
  - instructions and control structures (sequence, selection, iteration/repetition)
  - arithmetic, logical and conditional operators
  - graphical user interfaces (GUIs)
  - functions and methods
  - classes and objects

# Unit 3 Outcome 1 – Key knowledge

- purposes and features of naming conventions for solution elements (variables, interface controls, code structures), including:
  - Hungarian notation
  - camel casing
  - snake casing
- validation techniques for data, including:
  - existence checking
  - type checking
  - range checking

# Unit 3 Outcome 1 – Key knowledge

- purposes of internal documentation, including:
  - explaining and justifying data and code structures
  - code maintenance
  - placeholder comments for future development (stubs)
- algorithms for sorting and searching, including:
  - selection sort
  - quick sort
  - binary search
  - linear search

# Unit 3 Outcome 1 – Key knowledge

- types of errors, including:
  - syntax
  - logic
  - runtime (overflow, index out of range, type mismatch, divide by zero)
- debugging and testing techniques for checking modules function correctly, including:
  - use of breakpoints
  - use of debugging statements
  - construction of relevant test data
  - test cases comparing expected and actual output in testing tables.



# Unit 3 Outcome 1 – Key skills

- interpret solution requirements and designs
- use a range of data types, data structures and data sources
- use and justify appropriate features of an OOP language to develop working software modules
- develop and apply suitable naming conventions and validation techniques within modules
- document the functioning of modules using internal documentation
- develop and apply suitable debugging and testing techniques using appropriate test data.

# Unit 3 Outcome 1

## Contribution to final assessment

- School-assessed Coursework for Unit 3 will contribute 10 per cent to the study score.
- Total marks – 100

# Unit 3 Outcome 1

## Assessment task

In response to teacher-provided solution requirements and designs, develop four working modules with increasing complexity of programming skills.

- Module 1: Simple calculations using arithmetic, logical and conditional operators
- Module 2: Reading and writing files
- Module 3: Sorting and searching with functions or methods
- Module 4: Classes and objects

At least two modules must include a GUI.

All modules must include testing.

Task time allocated should be at least 8–14 lessons.

# Contact

- Phil Feain – Digital Technologies Curriculum Manager (VCAA)
- Ph: (03) 9059 5146
- Philip.Feain@education.vic.gov.au

© Victorian Curriculum and Assessment Authority (VCAA) 2024. Some elements in this presentation may be owned by third parties. VCAA presentations may be reproduced in accordance with the [VCAA Copyright Policy](#), and as permitted under the Copyright Act 1968. VCE is a registered trademark of the VCAA.

Authorised and published by the  
Victorian Curriculum and Assessment Authority

