Sample Weekly Planner
Unit 3: Software development

Teachers can be flexible in the way they approach and deliver the two areas of study in Units 3 and 4. Two possible approaches are outlined below, either sequential or simultaneous. All units in the VCE are constructed on the basis of at least 50 hours of scheduled classroom instruction. Time allocations are suggested for each area of study in the sample planner.

|  |
| --- |
| **Sequential approach** |
| **Area of study** | **Suggested time allocation (weeks)** |
| 1. Software development: programming | Weeks 1–8\*\* |
| 2. Software development: analysis and design\* | Weeks 9–16\*\* |
|  |
| **Simultaneous approach** |
| **Area of study** | **Suggested time allocation (weeks)** |
| 3. Software development: programming | Weeks 1–12\*\* |
| 4. Software development: analysis and design\* | Weeks 5–16\*\* |

\* Unit 3 Outcome 2 forms part of the School-assessed Task.

\*\* Please note that the duration of each area of study is indicative only.

The unit planner below represents a **sequential approach** to delivering Unit 3: Software development. It is a sample guide only and teachers are advised to consider their own contexts when implementing this unit and when developing learning activities. Consideration should be given to the student cohort and available resources. Teachers should modify this sample weekly planner according to relevant school events.

Teachers wishing to adopt the **simultaneous approach** (when a class is completing the SAT and working on a SAC) can modify this detailed planner accordingly.

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Unit and area of study** | **Topic/description** | **Learning activities** |
| **Area of Study 1: Software development: programming** |
| 1 | Unit 3 Area of Study 1 | Data types and structures:* characteristics of data types.

Programming practice:* programming language
* processing features, such as instructions and control structures (sequence)
* naming conventions
 | * Discuss the outcome, and inform students of Unit 3 Outcome 1 SAC dates and conditions, as per school guidelines.
* Identify and justify specific uses of types of data.
* Practise programming tasks
	+ development environment setup
	+ teacher-led walkthrough of the key features of the development environment
	+ student exploration of the development environment
	+ displaying messages to screen
	+ simple mathematical calculations and string operations.
* Compare similar algorithms or programs where one uses a consistently applied naming convention and the other does not.
 |
| 2 | Unit 3 Area of Study 1 | Solving problems, interpreting requirements, constraints and scopeValidation techniquesProgramming practice:* processing features, such as control structures (sequence, selection)
* validation techniques (code and control-based)
 | * Examine existing solutions to identify solution requirements.
* Examine examples of data validation in existing solutions.
* Practise development of writing problem statements and documenting problems.
* Practise programming tasks:
	+ programming decision-making (If/Else, If/Else If)
	+ develop a software module that demonstrates different validation techniques using code- and control-based approaches.
 |
| 3 | Unit 3 Area of Study 1 | Interpreting teacher-provided requirements and designsInternal documentationTesting techniques:* testing tables
* construction of test data

Programming practice:* processing features, such as control structures (sequence, selection and iteration)
* internal documentation
 | * Based on teacher-provided software requirements and designs, students practise identifying key features of designs, structure of design tools and the relationship between provided requirements and designs.
* Annotate key aspects of the interface from an existing software solution that is familiar to students.
* Practise programming tasks:
	+ programming decision-making (Switch statements)
	+ using loops (pre-test, post-test)
	+ internal documentation.
* Practise using appropriate internal documentation in software solutions that explains the intent of the module and justifies the use of particular processing features.
* Demonstrate and practise the construction of test data.
* Document all testing conducted with programming tasks this week, using test tables and debugging techniques.
 |
| 4 | Unit 3 Area of Study 1 | Data types and structures:* arrays, records
* associative arrays

Programming practice:* processing features, such as control structures (iteration)
* arrays and records

Introduce SAT to students  | * Introduce the SAT to students so they can research/explore an appropriate need or opportunity.
* Compare a range of data structures and discuss their applicability in a range of scenarios.
* Apply hashing functions to sample data sets.
* Students create their own hashing functions, which are then tested on sample data sets.
* Practise programming tasks:
	+ using loops (fixed loops)
	+ initialising and working with arrays and records using loops.

**Preparing students for the Unit 3 Outcome 2 SAT**Outline the SAT to students by going through the requirements of the task, administration and compliance within the task and the establishment of processes for authentication. |
| 5 | Unit 3 Area of Study 1 | Sorting algorithms including selection sort and quick sortSearching algorithms including binary and linear searchingProgramming practice:* functions and methods
 | * Role-play the algorithmic behaviours of sorting and searching algorithms.
* Contrast the algorithmic differences, complexity and efficiency of selection and quick sort.
* Class discussion involving the justification of the use of linear or binary search techniques in particular situations.
* Practise programming tasks:
	+ write simple functions and methods that can be used for everyday situations
	+ develop a module that performs a linear or binary search.
 |
| 6 | Unit 3 Area of Study 1 | Testing techniques:* trace tables

Programming practice:* classes
 | * Using teacher-provided algorithms, students trace the variables using a trace table.
* Practise programming tasks:
	+ write simple functions and methods that can be used for everyday situations
	+ develop a module that performs a linear or binary search.

**Preparing students for the Unit 3 Outcome 2 SAT** * Check in with students with regards to the selection of their focus for the SAT.
 |
| 7 | Unit 3 Area of Study 1 | Formatting and structural characteristics of files (TXT, CSV and XML)Programming practice:* file operations
* using XML
 | * Compare the formatting and structural characteristics of TXT, CSV and XML files.
* Practise programming tasks:
	+ develop a module that reads data from a plain-text, delimited or XML file (pick one) and displays it in some way.
	+ develop a module that writes data within an application to an XML file.
 |
| 8 | Unit 3 Area of Study 1 | **Unit 3 Outcome 1 SAC** | **Unit 3 Outcome 1 SAC modules**Interpret teacher-provided solution requirements and designs, and apply a range of functions and techniques using a programming language to develop and test working software modules.**Number of modules:** 3–5**Time frame:** 3–5 hours (total)**Task:** Modules to be developed as part of this task should use a range of requirements and design tools to provide students with several opportunities to meet the requirements of the outcome. Modules should be of increasing difficulty across the range of modules completed. |
| **Area of Study 2: Software development: analysis and design** |
| 9 | Unit 3 Area of Study 2 | Confirmation of SAT projectsAnalysis:* organisational and information system goals and objectives
* requirements, constraints and scope
* data collection techniques

Legislative requirements around privacy and ownership of data | * Compare data collection techniques and justify their use in specific situations.
* Identify organisational and information system goals in familiar organsiations.
* Analyse mini-case studies and identify solution requirements and constraints, and how these may impact on solution scope.
* Prepare mini-case studies based on existing familiar solutions.
* Explore how organisations meet (or have not met) legislative requirements around ownership and privacy of data using anecdotes and media articles.

**Preparing students for the Unit 3 Outcome 2 SAT**Students are to confirm the problem, need or opportunity to be addressed, in consultation with their teacher.  |
| 10 | Unit 3 Area of Study 2 | Analysis:* software requirements specification

Project management conceptsProject management using Gantt chartsRecording the progress of projects  | * Develop Gantt charts and project plans in relation to medium- to long-term activities/events/projects.
* Analyse and deconstruct sample software requirements specification documentation to identify typical items for inclusion.
* Prepare a strategy to develop a software requirements specification.
 |
| 11 | Unit 3 Area of Study 2 | Analytical tools:* context diagrams
* data flow diagrams
* use case diagrams

Students working on their SATSAT Authentication sessions | * Examine the structural elements of analytical tools, and how structural elements have similar/different functions within each tool.
* Analyse teacher-provided analytical tools and identify errors with conventions. Refine diagrams as necessary.
* Interpret case studies and complete analytical diagrams.

**Unit 3 Outcome 2 SAT Authentication** Students work on Criteria 1–5 during class time.Students meet with teacher to monitor progress and update the Administration Record Form. |
| 12 | Unit 3 Area of Study 2 | Software design tools:* data dictionaries
* mock-ups
* object descriptions

Students working on their SATSAT Authentication sessions | * Generate a range of software designs based on teacher-provided case studies.
* Reverse engineer existing solution interfaces, identifying key design elements and how these meet intended requirements.

**U3 O2 SAT Authentication** Students work on Criteria 1–5 during class time.Students meet with teacher to monitor progress and update the Administration Record Form. |

|  |  |  |  |
| --- | --- | --- | --- |
| 13 | Unit 3 Area of Study 2 | Software design tools:* Pseudocode

Developing evaluation criteriaGenerating and evaluating alternative design ideasStudents working on their SATSAT Authentication sessions | * Analyse the structure of evaluation criteria within different contexts (designs, solutions, development models, project plans).
* Generate a range of algorithm designs based on teacher-provided case studies.

**Unit 3 Outcome 2 SAT Authentication** Students work on Criteria 1–5 during class time.Students meet with teacher to monitor progress and update the Administration Record Form. |
| 14 | Unit 3 Area of Study 2 | Design factorsUser experiencesDevelopment model approachesStudents working on their SATSAT Authentication sessions | * Discuss how design factors and principles are addressed within existing familiar solutions.
* Conduct a debate about whether user authentication or encryption are necessary (or not) to consider in the design or development phase.
* Identify positive or negative user experiences, and analyse how these contribute to user satisfaction, familiarity and use of the software.
* Investigate development models and link to the PSM, highlighting key characteristics and advantages for use.

**Unit 3 Outcome 2 SAT Authentication** Students work on Criteria 1–5 during class time.Students meet with teacher to monitor progress and update the Administration Record Form. |
| 15 | Unit 3 Area of Study 2 | Students working on their SATSAT Authentication sessions | **Unit 3 Outcome 2 SAT Authentication** Students work on Criteria 1–5 during class time.Students meet with teacher to monitor progress and update the Administration Record Form. |
| 16 | Unit 3 Area of Study 2 | Students working on their SATSAT Authentication sessionsSAT Submission\* | **Unit 3 Outcome 2 SAT Authentication** Students work on Criteria 1–5 during class time.Students meet with teacher to monitor progress and update the Administration Record Form.**Unit 3 Outcome 2 SAT Submission**Students submit SAT Criteria 1–5 to be assessed against the performance descriptors provided by VCAA.  |

\*SAT Submission date/s should take into account the following:

* Internal school programs and key dates
* Sufficient time to assess and moderate student submissions
* Sufficient time to enter SAT criteria scores into VASS.