



NSW SOFTWARE ENGINEERING 11–12 STAGE 6 • NESA 2022 SYLLABUS

Year 12 Weekly Teaching Program

Scope & Sequence

Programming for the Web • Secure Software Architecture • Software Automation

Term 4, 2026 → Term 2, 2027 | 30 Weeks

Covers every Year 12 syllabus dot point for the three strands listed above.
The Software Engineering Project strand is delivered as an separate practical project and package.

Table of Contents

Purpose and Scope	3
Course Learning Outcomes (NESA)	3
Term Overview	4
Full 30-Week Scope and Sequence	5
Programming for the Web — Weeks 1–10	5
Secure Software Architecture — Weeks 11–22	6
Software Automation — Weeks 23–30	7
Additional Assessment Resources	9
Acknowledgements & Third-Party Licensing	10

Purpose and Scope

This program delivers the three Year 12 content strands of the Software Engineering 11–12 Syllabus (NESA, 2022) — Programming for the Web, Secure Software Architecture and Software Automation — as a sequence of 30 weekly units across three terms. The Software Engineering Project strand is delivered separately as an ongoing practical project (see NESA syllabus pp.27–29) and is not included in this weekly booklet series.

Every syllabus dot point for the three strands listed above has been mapped to a specific week below. Each week produces four booklets:

- Booklet 1 — Student Teaching Booklet (content, worked examples, in-class practice questions)
- Booklet 2 — Teacher Teaching Booklet (same content + full worked answers + extended teaching notes)
- Booklet 3 — Student Homework Booklet (independent consolidation questions)
- Booklet 4 — Teacher Homework Answer Booklet (full answers + marking criteria)

Course Learning Outcomes (NESA)

By the end of Year 12, a student:

Code	Outcome
SE-12-01	justifies methods used to plan, develop and engineer software solutions
SE-12-02	applies structural elements to develop programming code
SE-12-03	analyses how current hardware, software and emerging technologies influence the development of software engineering solutions
SE-12-04	evaluates practices to safely and securely collect, use and store data
SE-12-05	explains the social, ethical and legal implications of software engineering on the individual, society and the environment
SE-12-06	justifies the selection and use of tools and resources to design, develop, manage and evaluate software
SE-12-07	designs, develops and implements safe and secure programming solutions
SE-12-08	tests and evaluates language structures to refine code
SE-12-09	applies methods to manage and document the development of a software project (Software Engineering Project strand)

Each weekly unit below states which of these outcomes it targets. Across the 30 weeks, SE-12-01 to SE-12-08 are addressed multiple times through the three strands; SE-12-09 is developed primarily through the separate Software Engineering Project strand, with light touchpoints in weeks covering documentation, version control and collaborative practice (e.g. Weeks 8, 9, 21).

Term Overview

Term	Calendar dates	Weeks	Strand covered
Term 4, 2026	Tuesday 13 October - Thursday 17 December 2026	1-10	Programming for the Web
Term 1, 2027	Thursday 28 January - Friday 9 April 2027 (plus Weeks 21-22 may run into early Term 2)	11-22	Secure Software Architecture
Term 2, 2027	Monday 26 April - Friday 2 July 2027	23-30	Software Automation

Note: term dates are the official 2026–2027 NSW Department of Education calendar dates (Eastern Division) at time of writing. Confirm against your school’s calendar and adjust for pupil-free days, assessment blocks and public holidays — the week numbers are curriculum weeks, not fixed calendar weeks, so Weeks 21–22 may run slightly past the nominal Term 1 end date into early Term 2 depending on how each term actually falls.

Full 30-Week Scope and Sequence

Programming for the Web

Week 1 — Applications of Web Programming

Textbook reference: Section 11.1 | Outcomes: SE-12-01, SE-12-03, SE-12-06

- Explore the applications of web programming, including: interactive website/webpages, e-commerce, progressive web apps (PWAs)

Week 2 — How Data Moves on the Internet

Textbook reference: Section 11.2 | Outcomes: SE-12-03

- Investigate and practise how data is transferred on the internet, including: data packets, IP addresses including IPv4, domain name systems (DNS)

Week 3 — Web Protocols, Ports and Secure Transport

Textbook reference: Section 11.3 | Outcomes: SE-12-03, SE-12-04, SE-12-07

- Investigate and describe the function of web protocols and their ports, including HTTP, HTTPS, TCP/IP, DNS, FTP/SFTP, SSL/TLS, SMTP/POP3/IMAP
- Explain the processes for securing the web, including SSL certificates, encryption algorithms and keys, plain text/cipher text, authentication and authorisation, hash values, digital signatures

Week 4 — Big Data and Web Architectures

Textbook reference: Section 11.4 | Outcomes: SE-12-03, SE-12-06

- Investigate the effect of big data on web architecture, including data mining, metadata, streaming service management

Week 5 — The Role of the W3C and Web Standards

Textbook reference: Section 12.1 | Outcomes: SE-12-01, SE-12-05

- Investigate and explain the role of the W3C in the development of applications for the web, including WAI, internationalisation, web security, privacy, machine-readable data

Week 6 — Model Elements of a Web Development System

Textbook reference: Section 12.2 | Outcomes: SE-12-02, SE-12-06

- Model elements that form a web development system, including client-side (front-end), server-side (back-end), interfacing with SQL/non-SQL databases

Week 7 — Browser Influence, Dev Tools, CSS and UI/UX Principles

Textbook reference: Section 12.3 + 12.4 | Outcomes: SE-12-01, SE-12-06

- Explore and explain the influence of a web browser on web development, including dev tools
- Investigate CSS and its impact on the design of a web application, including consistency of appearance, flexibility with browsers/display devices, CSS maintenance tools

Week 8 — Front-End Libraries/Frameworks and Open-Source/CMS

Textbook reference: Section 12.5 + 12.6 | Outcomes: SE-12-01, SE-12-05, SE-12-06

- Investigate the reasons for version control
- Explore the types and significance of code libraries for front-end web development, including frameworks, template engines, predesigned CSS classes

- Explain the use and development of open-source software in relation to web development
- Research, experiment with and evaluate the prevalence and use of web content management systems (CMS)

Week 9 — Back-End Request Flow and Databases (SQL/ORM)

Textbook reference: Section 13.1 + 13.2 | Outcomes: SE-12-02, SE-12-06, SE-12-07, SE-12-08

- Assess the contribution of back-end web development to the success of a web application; observe and describe the back-end process used to manage a web request (web server software, web framework, objects, libraries, databases)
- Apply a web-based database and construct a script that executes SQL, including selecting fields, GROUP BY, common SQL queries, WHERE, table joins
- Compare Object-Relational Mapping (ORM) to SQL

Week 10 — Performance/Page-Load Management and Designing a PWA

Textbook reference: Section 13.3 + 13.4 | Outcomes: SE-12-01, SE-12-03, SE-12-06

- Investigate methods to support and manage the load times of web pages/applications
- Design, develop and implement a progressive web app (PWA), including UI/UX principles of font, colour, audio, video, navigation, and a UI that considers accessibility and inclusivity
- Describe how collaborative work practices between front-end and back-end developers improve development of a web solution

Secure Software Architecture

Week 11 — The Business Case for Security

Textbook reference: Section 14.1 | Outcomes: SE-12-01, SE-12-05

- Describe the benefits of developing secure software, including data protection, minimising cyber attacks and vulnerabilities

Week 12 — Integrating Security into the SDLC

Textbook reference: Section 14.2 | Outcomes: SE-12-01, SE-12-02, SE-12-07

- Interpret and apply fundamental software development steps to develop secure code, including requirements definition, determining specifications, design, development, integration, testing and debugging, installation, maintenance

Week 13 — User-Centred Security Design

Textbook reference: Section 14.3 | Outcomes: SE-12-01, SE-12-05

- Describe how the capabilities and experience of end users influence the secure design features of software

Week 14 — Security Fundamentals: CIA Triad and AAA

Textbook reference: Section 15.1 | Outcomes: SE-12-04, SE-12-07

- Explore fundamental software design security concepts when developing programming code, including confidentiality, integrity, availability, authentication, authorisation, accountability

Week 15 — Cryptography and Data Protection

Textbook reference: Section 15.2 | Outcomes: SE-12-04, SE-12-07

- Use and explain the contribution of cryptography and sandboxing to the 'security by design' approach in the development of software solutions

Week 16 — Input Validation and Sanitisation

Textbook reference: Section 16.1 | Outcomes: SE-12-07, SE-12-08

- Design, develop and implement code using defensive data input handling practices, including input validation, sanitisation and error handling

Week 17 — Privacy by Design

Textbook reference: Section 16.2 | Outcomes: SE-12-04, SE-12-05

- Use and explain the 'privacy by design' approach in the development of software solutions, including proactive not reactive approach, embed privacy into design, respect for user privacy

Week 18 — Secure API Design

Textbook reference: Section 17.1 | Outcomes: SE-12-02, SE-12-06, SE-12-07

- Design, develop and implement a safe application programming interface (API) to minimise software vulnerabilities

Week 19 — Secure Execution and Resource Management

Textbook reference: Section 17.2 | Outcomes: SE-12-07, SE-12-08

- Design, develop and implement code considering efficient execution for the user, including memory management, session management, exception management
- Design, develop and implement secure code to protect user file and hardware vulnerabilities from file attacks and side-channel attacks

Week 20 — Security Testing Fundamentals and Web Application Vulnerabilities

Textbook reference: Section 18.1 + 18.2 | Outcomes: SE-12-07, SE-12-08

- Test and evaluate the security and resilience of software by determining vulnerabilities, hardening systems, handling breaches, maintaining business continuity and conducting disaster recovery
- Apply and evaluate strategies used by software developers to manage the security of programming code, including code review, SAST, DAST, vulnerability assessment, penetration testing
- Design, develop and implement secure code to minimise vulnerabilities in user action controls, including broken authentication/session management, XSS, CSRF, invalid forwarding/redirecting, race conditions

Week 21 — Security in Development Teams and Enterprise Security Benefits

Textbook reference: Section 19.1 + 19.2 | Outcomes: SE-12-01, SE-12-05, SE-12-06

- Apply and describe the benefits of collaboration to develop safe and secure software, including considering various points of view, delegating tasks based on expertise, quality of the solution
- Investigate and explain the benefits to an enterprise of the implementation of safe and secure development practices, including improved products/services, influence on future software development, improved work practices, productivity, business interactivity

Week 22 — Security Ethics, Legal Ramifications and Evaluating Security Programs

Textbook reference: Section 19.3 + 19.4 | Outcomes: SE-12-01, SE-12-05, SE-12-06

- Evaluate the social, ethical and legal issues and ramifications that affect people and enterprises resulting from the development and implementation of safe and secure software, including employment, data security, privacy, copyright, intellectual property, digital disruption

Software Automation

Week 23 — What is AI vs ML (and where RPA/BPA fit)

Textbook reference: Section 20.1 | Outcomes: SE-12-03, SE-12-06

- Investigate how ML supports automation through the use of DevOps, robotic process automation (RPA) and business process automation (BPA)

Week 24 — ML Training Models: Supervised, Unsupervised, Semi-Supervised, Reinforcement

Textbook reference: Section 20.2 | Outcomes: SE-12-06

- Explore models of training ML, including supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning

Week 25 — Common ML Applications

Textbook reference: Section 20.3 | Outcomes: SE-12-03, SE-12-06

- Investigate common applications of key ML algorithms, including data analysis and forecasting, virtual personal assistants, image recognition

Week 26 — Design Models: Decision Trees and Neural Networks

Textbook reference: Section 20.4 | Outcomes: SE-12-06

- Research models used by software engineers to design and analyse ML, including decision trees, neural networks
- Describe types of algorithms associated with ML, including linear regression, logistic regression, K-nearest neighbour

Week 27 — RPA and BPA: Practical Considerations

Textbook reference: Section 20.5 | Outcomes: SE-12-01, SE-12-05, SE-12-06

- Investigate how ML supports automation through DevOps, RPA and BPA (practical/ethical/security application)

Week 28 — Programming for Automation: Regression Models and Neural Networks in Python

Textbook reference: Section 21.1 + 21.2 | Outcomes: SE-12-02, SE-12-06

- Design, develop and apply ML regression models using an OOP to predict numeric values, including linear regression, polynomial regression, logistic regression
- Apply neural network models using an OOP to make predictions

Week 29 — Significance and Impact of ML/AI, and Human Behaviour Patterns

Textbook reference: Section 22.1 + 22.2 | Outcomes: SE-12-05

- Assess the impact of automation on the individual, society and the environment, including safety of workers, people with disability, nature and skills required for employment, production efficiency/waste/environment, economy and distribution of wealth
- Explore by implementation how patterns in human behaviour influence ML and AI software development, including psychological responses, acute stress response, cultural protocols, belief systems

Week 30 — Bias in Datasets and Models, plus Course Consolidation

Textbook reference: Section 22.3 + Course Revision | Outcomes: SE-12-05

- Investigate the effect of human and dataset source bias in the development of ML and AI solutions

Additional Assessment Resources

Beyond the 30 weekly units, this program also includes cumulative assessment resources, each with a student paper and a separate teacher marking-guidelines document.

Topic Tests (end-of-strand mock exams)

Resource	Coverage	Marks
Topic Test 1 — Programming for the Web	Weeks 1–10	48
Topic Test 2 — Secure Software Architecture	Weeks 11–22	65
Topic Test 3 — Software Automation	Weeks 23–30	47

Trial HSC Examination

A comprehensive practice examination (79 marks, 140 minutes working time + 10 minutes reading) spanning all three strands proportionally, including a final synthesis question that draws on all three strands at once. Calibrated to match the real HSC Software Engineering exam, which is delivered online and is worth 80 marks over 2 hours 30 minutes including 10 minutes reading time.

Software Engineering Project Guide

A standalone guide to the fourth syllabus strand (Software Engineering Project) — covering every syllabus requirement across Identifying & Defining, Research & Planning, Producing & Implementing, and Testing & Evaluating, plus practical guidance on producing strong evidence, a self-assessment checklist, and common pitfalls to avoid.

A note on question sourcing

- Multiple-choice and written questions in the topic tests and trial exam draw on the same content covered in each week's practice and homework booklets, reassembled into cumulative papers — this keeps practice consistent with what was actually taught while giving students genuine exam experience under timed conditions.

Acknowledgements & Third-Party Licensing

Some technical explanations, worked examples and structural approaches in this teaching program were informed by and adapted from the Software Engineering HSC Textbook (github.com/Eatham532/Software-Engineering-HSC-Textbook), an open-source educational resource. That project is released under the MIT License, reproduced in full below as required by its terms.

All content in this program has been independently written, restructured and substantially expanded for this course — including original worked examples, practice/homework questions, marking criteria, code samples, topic tests, a full trial HSC examination and the Software Engineering Project guide — but the underlying MIT licence requires this notice to accompany any use of the source material it covers.

MIT License

Copyright 2025 Eatham532

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Also acknowledged

- NSW Education Standards Authority (NESA) — the Software Engineering 11–12 Syllabus (2022) content descriptors and outcomes.
- This program is an independent teaching resource prepared for TH Software Tuition and is not affiliated with or endorsed by NESA or the Eatham532 Software Engineering HSC Textbook project.