Sample Weekly Planner   
Unit 1: Applied computing

The unit planner below represents one approach to delivering Unit 1: Applied computing. 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.

|  |  |
| --- | --- |
| **Area of Study** | **Suggested time allocation (Weeks)** |
| 1. Data analysis | Weeks 1–8\*\* |
| 1. Programming | Weeks 9–16\*\* |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Week** | **Unit and area of study** | **Topic/description** | **Learning activities** |
| **Area of Study 1: Data analysis** | | | |
| 1 | Unit 1  Area of Study 1 | Data and information:   * types, purposes, characteristics and quality * sources, methods and techniques for acquiring data * interpretation and communication * legal and ethical data collection | * Preliminary class looks at how organisations collect and use data; with consideration of legal and ethical issues involving the collection of data. * Investigate the methods of collecting data, such as surveys. * Class explores some of the big data repositories, such as: * openNASA: exploration of NASA’s data portal (e.g. data evidence for climate change) * Australian Bureau of Statistics: exploration of ABS data sources (e.g. safety and justice data). |
| 2 | Unit 1  Area of Study 1 | Software tools – Excel:   * structural characteristics | Students complete a sequence of Excel activities to develop their knowledge and skills:   * entering data into cells * creating simple formulas * formatting data and cells * sorting * using functions (sum, average, max, min, count, count if) * creating lookup tables (hlookup, vlookup) * creating graphs from sample data * adding lines of best fit.   Note: student ability levels will vary. |
| 3 | Unit 1  Area of Study 1 | Spreadsheets and data visualisations:   * design tools for representing functionality and appearance of spreadsheets and data visualisations * formats and conventions | * Class examines a range of data visualisations (provided by the teacher) and their purpose, which leads to discussion of formats and conventions used. * Students create a basic data visualisation from sample data provided by the teacher. * Students are given stimulus material, such as an extract from a newspaper article. The purpose of giving them the article is to engage the students in further research that relates to an authentic issue reported in the media. Suitable relevant examples could include: Australia’s power consumption; crime statistics or sports data. Students collect the data, import it into Excel and create a data visualisation. |
| 4 | Unit 1  Area of Study 1 | Databases and database management systems:   * design tools for representing functionality and appearance of databases * formats and conventions | * The importance of naming conventions is introduced to the class, including the use of meaningful names when creating a database, tables and other objects. * Students create a data dictionary to define the contents and format of a database. * Normalisation of data is discussed, with students completing activities involving placing non-normalised data into 1st normal form, 2nd normal form and 3rd normal form. |
| 5 | Unit 1  Area of Study 1 | Databases and database management systems:   * design tools for representing functionality and appearance of databases * formats and conventions * protecting data | * Students complete a series of activities to help develop their skills and knowledge in using databases and database management systems: * create a database, add permissions * create, modify and delete tables * create primary and foreign keys * run queries to insert, update, display and deleting data * run queries across multiple tables * import data into the database. |
| 6 | Unit 1  Area of Study 1 | Research task:   * teacher-provided research question * data manipulation and validation | * Teacher creates a research question as an information problem. Project scope and constraints are provided to students. Potential sources of data are provided to students. * Students identify a list of data requirements, including data type. Additional and/or alternative sources of data are identified by the students as part of their research. |
| 7 | Unit 1  Area of Study 1 | Research task:   * teacher-provided analysis * data visualisations | * Students collect and validate data (note: any qualitative data is coded into a quantitative form). * Students use a design tool to create samples of their spreadsheet and data visualisations; preferred designs are selected. * Data is organised and imported into Excel, and data visualisations created. |
| 8 | Unit 1  Area of Study 1 | Privacy and ethical issues | * Students analyse a case study provided by the teacher, which explores data privacy and ethical issues. A suitable case study could be framed around a contemporary issue, such as social media’s use of personal data; the US government wanting access to the iPhones of terrorist suspects and Apple’s refusal; or Google Street View being brought to remote indigenous communities. |
| **Area of Study 2: Programming** | | | |
| 9 | Unit 1  Area of Study 2 | Design tools in programming:   * pseudocode * data dictionaries * mock-ups | * Students investigate the concepts of algorithms and pseudocode. They review the importance and characteristics of data dictionaries and mock-ups. * Activities undertaken by students include: writing pseudocode, debugging teacher-provided pseudocode, creating test data for pseudocode. * Download and install the programming software that students will use, including configuring any IDE (Integrated Development Environment). |
| 10 | Unit 1  Area of Study 2 | Programming fundamentals:   * characteristics of data types * types of data structures * naming conventions * processing features * internal documentation | * Students investigate simple programs to develop their understanding and knowledge of the concepts of programming, and the syntax of the programming language being used. Teacher provides guided instruction. * Students convert an algorithm from pseudocode into their high-level programming language. * Sample programming activities that could be completed to assist in developing students’ knowledge and skills include: * initialising variables * assigning values to variables * performing calculations * printing the stored value of a variable * using control functions, including sequence, selection (e.g. If-Then-Else) and iteration (e.g. While loops) * string-handling operations such as substring and concatenation. * Emphasis is on good programming practice, where students adhere to standards including good variable names, internal documentation, and indenting. |
| 11 | Unit 1  Area of Study 2 | Programming fundamentals:   * processing features such as nested selection and nested loops * writing programs to solve a problem   Program testing and debugging | * Students continue to develop their knowledge and skills through writing programs that use: * nested selection and nested loops * arrays * pass values to a function * receive a value back from a function. * Other programs are written in response to teacher-provided problems, such as: * calculating the volume of a sphere based on user-input * calculating how much money will be required for a company’s payroll, and the amount of tax that needs to be paid. * Students develop test tables, and use tracing as a means of debugging programs, either by hand or within the code itself. |
| 12 | Unit 1  Area of Study 2 | Programming fundamentals:   * data and file management | * Students learn (and practise) writing programs that can: * open and close a file * write data to a file * read data from a file. |

|  |  |  |  |
| --- | --- | --- | --- |
| 13 | Unit 1  Area of Study 2 | Software development:   * create a working software solution (from teacher-provided solution requirements) | * Teacher provides students with a problem requiring a software solution. Project scope and constraints are provided to students. * Sample requirements for the programming solution include: * an authentication routine that uses a username and password * use of a single-dimensional array * being able to read and write to a file * exception handling * creating a password-protected file. |
| 14 | Unit 1  Area of Study 2 | Software development:   * create a working software solution (from teacher-provided solution requirements) * testing and debugging | * Students continue working on their software solution. * They test all features of the solution’s functionality and use a test table that they have created. * Correct all errors identified and modify code to handle any exceptions not previously captured. |
| 15 | Unit 1  Area of Study 2 | Software development:   * evaluating the efficiency and effectiveness of a software solution | Students evaluate the efficiency and effectiveness of their software solution. |
| 16 | Unit 1  Area of Study 2 | Unit 1 revision |  |