Declarative and Procedural Knowledge

Year 3

Contents

Introduction

The Declarative and Procedural Knowledge documents are designed to support teachers in understanding the intended learning outcomes of each unit. They outline the specific knowledge and skills that children should acquire and demonstrate by the end of their learning.

- Declarative Knowledge sets out what children will ***know***. This includes facts, concepts, definitions, and key ideas that form the foundation of the unit.
- Procedural Knowledge sets out what children will ***be able to do***. This focuses on the skills and processes children should develop and apply when using technology.

These documents are used to:

- Provide teachers with a clear overview of learning expectations for each unit.
- Ensure consistency of teaching and progression of knowledge and skills across year groups.
- Support planning, teaching, and assessment by highlighting the essential outcomes to focus on.
- Reinforce the balance between understanding (knowing) and application (doing) in computing.

This document aims to help teachers see the bigger picture of what children will learn, how they will apply it, and how these elements connect across the computing curriculum.

Introduction to Purple Mash

National Curriculum Links	**Dominant objectives for this unit:** Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.

Declarative - By the end of the unit the children **will know that**:	**Procedural** - By the end of the unit the children **will know how to**:
• It is important to log in to a site, the importance of keeping passwords safe and the need to log out at the end of a session.	• Access Purple Mash from home and school. • Log out of Purple Mash. • Give reasons why it is important to keep a password safe and not share it with other people.
• An avatar is a virtual representation of a person suitable for use online.	• Make and edit their own avatar.
• The 2Do system is used to set work for children within Purple Mash.	• Open 2Dos. • Save 2Dos. • Hand in 2Dos and communicate with their teacher via the 2Do.
• Online sites have a main page called the homepage.	• Access the Purple Mash homepage when on the site.
• Online sites often use an alert system to communicate with the user.	• Access alerts within Purple Mash.
• To move to a different activity in Purple Mash, you must close the current activity.	• Close activities in Purple Mash.
• Many online sites, including Purple Mash, have an area for an individual's work that is accessible only to the individual (and in Purple Mash to their teacher as well).	• Access their work area. • Save work in their work area. • Locate and open work they have done previously in their work folder.
• To access Purple Mash programs, you use the Tools area.	• Open a specified tool.
• You can access non-visible parts of a screen using scrolling.	• Scroll up and down and from side to side where applicable.

Email

National Curriculum Links	**Dominant objectives for this unit:** • Use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact. • Understand computer networks, including the Internet; how they can provide multiple services, such as the World Wide Web; and the opportunities they offer for communication and collaboration.

Declarative - By the end of the unit the children **will know that**:	**Procedural** - By the end of the unit the children **will know how to**:
• There are different methods of communication and they each have strengths and weaknesses.	• Explain the advantages and disadvantages of different communication methods. • Choose an appropriate communication method for a task.
• Emails are a form of digital communication.	• Make use of 2Email to communicate within school. • Recognise the differences between digital and non-digital communication methods. • Explain different digital communication methods such as instant messaging, email and video calls.
• Emails can be sent and received almost instantly to anyone with an email address.	• Identify the advantages of instant communication. • Explain the differences between email and traditional communication methods which are not instant.
• Common features of email software are the inbox, the 'To' address field, the sender email address, the subject, the message text, and the compose and reply functions.	• Identify the common features of the email screen. • Make use of these common features of email software to communicate digitally.
• Alerts can be used to notify a person that they have unread email.	• Check alerts for new messages and respond to these.
• Address books can be saved in the email software. This provides a convenient way to send emails without typing the full email address each time.	• Use the address book within 2Email to find contacts. • Send emails to multiple contacts using the address book.
• Pictures, documents and other file types can be attached to emails.	• Identify the attachment icon. • Select files to attach to an email and send. • Be cautious of emails that have attachments. • Discuss the advantages and disadvantages of being able to send attachments with emails.

• There are risks related to use of email.	• Recognise a concerning email/contact. • Identify who a trusted contact is. • Report any concern to a trusted adult and use the report to teacher feature in 2Email. • Consider consent when sharing content digitally. • Recognise personal and private information and know what is not appropriate to share with a recipient.

Route Planners

National Curriculum Links	**Dominant objectives for this unit:** • Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. • Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Declarative - By the end of the unit the children **will know that**:	**Procedural** - By the end of the unit the children **will know how to**:
• The combination of a direction and a distance is known as a command in 2Go.	• Input commands in 2Go.
• Commands can be input into 2Go to control the movement of a screen turtle in four directions.	• Input purposeful commands in 2Go to make the turtle move in a particular direction towards a goal.
• The turtle can be set to rotate by angles of 90° or 45° both clockwise and anticlockwise.	• Input commands that rotate the turtle to face the desired direction for movement using both 90° and 45° angles.
• Planning a route is important to ensure the correct commands are input.	• Use techniques such as finger movements to plan a route.
• A list of instructions for a route is called an algorithm.	• Plan the route by first writing the algorithm and then inputting the code commands.
• Routes can be programmed to perform more than one command in a sequence.	• Input several commands into a sequential algorithm layout and run this code to move the turtle along the programmed route. • Reset the turtle to the starting position to re-run the code.
• Routes can be programmed to repeat a sequence of commands a set number of times.	• Use the repeat algorithm layout for 2Go commands. • Anticipate the effect of the repeat and use logical reasoning to decide upon when this layout is useful to accomplish a task.
• Errors (bugs) occur because commands may have been input incorrectly. • Fixing the errors is called debugging.	• Make logical attempts to debug code for routes. • Reset, debug and re-run the code to test routes.

Branching Databases

National Curriculum Links	**Dominant objectives for this unit:** • Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.

Declarative - By the end of the unit the children **will know that**:	**Procedural** - By the end of the unit the children **will know how to**:
• A database is a collection of data organised in a way that it can be searched, and information found easily.	• Explain what a database is. • Provide examples of common uses of a database such as the school's attendance database.
• A binary question is one that can be answered with 'yes' or 'no'.	• Explain the differences between a binary question and an open question.
• Branching databases are structured using binary choices.	• Identify binary questions that could be used to sort items. • Sort items according to whether the answer the question is yes or no.
• A good binary question splits a group roughly in half, half with the answer yes, half with the answer no.	• Choose binary questions that split a group of objects into half.
• When using a binary database, the questions eliminate data until just one record is left, the item can then be identified.	• Use a binary database in 2Question to identify items.
• Branching databases can be created using programs such as 2Question.	• Add records to a 2Question database then sort the records using binary questioning to create the database.
• It is important to test and debug when creating branching databases so that it works as intended.	• Work through all routes of a branching database to test whether it works as intended. • Identify and fix bugs in a database.

Touch Typing

National Curriculum Links	**Dominant objectives for this unit:** • Select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information.

Declarative - By the end of the unit the children **will know that**:	**Procedural** - By the end of the unit the children **will know how to**:
• Typing is the action or skill of writing something by means of a keyboard (physical or virtual).	• Use basic touch-typing skills to become quicker and more efficient with typing.
• It is important to have a good posture when typing to ensure their bodies remain comfortable and don't get sore.	• Check that posture is correct when typing. • Position equipment correctly so that eyes are level with monitor. • Place feet on the floor. • Position wrists so that they are not touching anything when typing.
• A keyboard is a device with a set of keys (buttons) that you press to type letters, numbers, punctuation and symbols into a computer. It can come in different shapes and sizes.	• Use a range of different keyboards to type letters, numbers, punctuation and symbols.
• To be efficient at typing, hands should be positioned correctly on a keyboard and the left and right hands should work independently of each other.	• Use the left hand to type letters. • Use the right hand to type letters. • Locate the ridges on keys F and J to find where the index fingers should rest on the keys. • Position the left and right hands correctly. • Build up to combing left and right-hand use to type words.
• Home, top and bottom row keys are areas on a keyboard where specific keys are located.	• Locate the home, top and bottom keys. • Open activities in 2Type referencing the keyboard guide to support recognition of using the correct key when typing letters.
• Numbers are located on the Number Row at the top of the keyboard.	• Type numbers using the correct hand and fingers.
• The shift key is used alongside another key to type capital letters and different punctuation marks and symbols.	• Use the shift key by pressing the key on either side of the keyboard with their little finger.
• The space bar is pressed with the thumb and types a space between letters.	• Press the space bar efficiently to form spaces between words.

micro:bits

National Curriculum Links	**Dominant objectives for this unit:** • Design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts. • Use sequence, selection, and repetition in programs; work with variables and various forms of input and output. • Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Declarative - By the end of the unit the students **will know that**:	**Procedural** – By the end of the unit the students **will know how to**:
• A micro:bit is a tiny computer which needs instructions in code to make it work.	• Explain that a micro:bit is a piece of hardware that can have code created for it that makes use of its inputs and outputs. • Recognise and locate key hardware components on the micro:bit such as its display, speaker and accelerometer.
• A micro:bit can produce outputs.	• Identify and use code blocks that produce outputs. • Use display text, show LEDs and play music outputs in program to meet specific intentions. • Code a micro:bit to make different outputs happen depending on different inputs.
• A micro:bit can receive inputs.	• Identify and use code blocks that are associated with receiving inputs. • Use event commands such as 'when micro:bit button' and 'when gesture' in programs to meet specific intentions. • Make a program that requires inputs (event commands as above) that produce an output.
• Code from the coding environment can be transferred onto a micro:bit.	• Use the simulator within the Freecode micro:bit environment to test code before transferring to micro:bit. • Use the transfer feature to move code to a micro:bit.
• The order (sequence) of instructions is important when coding.	• Recognise how the order of code is essential in order to meet a program's intentions. • Use sleep command within a sequence of code to temporarily pause a program e.g. when creating an animation effect using LEDs. • Use the repeat forever command within a sequence of code to produce an infinite looping sequencing such as a beating heart animation.