Curriculum Knowledge Map 2024-25

Year 8

Unit 1: The Power of 0’s and 1’s: Unlocking the Secrets of Computing

Rationale:

This introductory Year 8 computing unit aims to demystify the inner workings of computers and the digital world. By exploring the fundamental concept of binary code and its role in everything from cybersecurity to data representation, students will gain a foundational understanding of how computers operate and communicate. Through hands-on activities and investigations, students will develop essential computational thinking skills, paving the way for more advanced programming and problem-solving in future units

Declarative

What should they know?

Definitions:

- **Cybersecurity**: The practice of protecting computers, networks, and data from unauthorized access, theft, damage, or disruption.
- **Cyber Threats**: Malicious activities that target computer systems or networks, including viruses, malware, phishing, hacking, and DDoS attacks.
- **Vulnerability**: A weakness in a computer system, network, or software that can be exploited by attackers.
- **Operating System**: Software that manages computer hardware and software resources and provides common services for computer programs.
- **Computer Network**: A set of interconnected computers that can communicate and share resources.
- **Encryption**: A method of converting data into a coded form to prevent unauthorized access.
- **Binary Code**: A system of representing data using only two digits: 0 and 1.
- **CPU** (**Central Processing Unit**): The "brain" of the computer, responsible for executing instructions.
- **RAM** (**Random Access Memory**): Temporary storage for data and instructions while the computer is running.
- **Storage Devices**: Hardware that stores data permanently (e.g., hard disk drives, solid-state drives).
- **Fetch-Decode-Execute Cycle**: The basic operation cycle of a CPU.
- **Logic Gates**: Electronic circuits that perform basic logical operations and are the building blocks of computer processors.
- **Algorithm**: A step-by-step set of instructions designed to solve a problem or complete a task.

Facts:

- Cyber threats are constantly evolving and can have serious consequences.
- Strong passwords, software updates, and safe online habits are essential for protection.
- Networks enable communication and resource sharing between computers.
- Encryption helps protect data transmitted over networks.
- Computers use binary code to represent and process all types of data.
- The CPU follows a fetch-decode-execute cycle to process instructions.
- Logic gates are used to make decisions and perform calculations in computers.
- Algorithms are used to solve problems in both everyday life and computing.

Procedural

What should they be able to do?

Skills

- Identify common cyber threats and vulnerabilities.
- Describe basic cybersecurity measures for individuals and organizations.
- Explain the role of operating systems and software in security.
- Differentiate between types of computer networks.
- Explain how data is transmitted over networks using packets and protocols.
- Describe the function of common hardware components (CPU, RAM, storage).
- Explain how the CPU fetches, decodes, and executes instructions in binary code.
- Interpret truth tables for logic gates (AND, OR, NOT).
- Identify how binary code is used to represent text, images, and sound.
- Define and provide examples of algorithms in everyday life and computing.

Disciplinary Literacy

(Tier 3 Vocab)

- Cybersecurity
- Malware
- Phishing
- Hacking
- Firewall
- Operating System (OS)
- Computer Network
- LAN (Local Area Network)
- WAN (Wide Area Network)
- Encryption
- Binary Code
- AND / OR / NOT Gates
- CPU (Central Processing Unit)
- RAM (Random Access Memory)
- Storage
- Fetch-Decode-Execute Cycle
- Logic Gates

Assessment

Formative Assessment:

- **Do Now Recaps**: Daily "Do Now" activities will assess prior knowledge and understanding from previous lessons.
- **End of Lesson MS Forms**: Formative quizzes administered through Microsoft Forms will gauge student comprehension of concepts covered during the lesson.
- **Hardware Diagram:** Diagram labelling of hardware components and their functions.
- **Binary Code Practical:** Practical exercises demonstrating binary code conversion and interpretation.

Summative Assessment:

- **Algorithm Design Challenge:** Summative project where students create a simple algorithm or program that demonstrates their understanding of the unit's concepts.
- **Unit Assessment:** Hardware Architecture

Curriculum Knowledge Map 2024-25

Year 8	Unit 2: Computational Thinking: Abstractions, Algorithms, Simulations & Models		
Rationale:	This unit bridges the gap between theory and practice by applying the knowledge gained in Units 1 and 2 to the creation of a simple game. Using the game (Ms) Pac-Man as a context, students will explore how computational abstractions are used to model game elements and how programming constructs (sequence, selection, iteration) bring these elements to life. This unit reinforces their understanding of key concepts, introduces them to the world of programming, and sparks their creativity.		
Declarative *What should they know?*	Definitions: • **Computational Thinking**: A problem-solving approach that involves decomposition, pattern recognition, abstraction, and algorithm design. • **Algorithm**: A step-by-step procedure or set of rules for solving a problem or accomplishing a task. • **Flowchart**: A graphical representation of an algorithm using symbols to show the sequence of steps and decision points. • **Pseudocode**: A plain language description of an algorithm, not tied to any specific programming language. • **Search Algorithm**: A methodical approach to finding a specific item in a collection of data. • **Cipher**: A method of transforming a message to conceal its meaning. • **Efficiency**: How well an algorithm performs in terms of time (how long it takes to run) and space (how much memory it uses). • **Logical Reasoning**: The process of using rational, systematic steps, based on mathematical procedures, to arrive at a conclusion. • **Iteration**: The repetition of a process or set of instructions in a computer program. • **Encryption**: The process of converting information or data into a code, especially to prevent unauthorized access. • **Decryption**: The process of converting encrypted data or information back into its original form.	Facts: • Computational thinking skills are valuable in various fields beyond computer science. • There are different types of maze-solving algorithms, each with its own advantages and disadvantages. • Algorithms can be used to encrypt and decrypt messages. • Logic puzzles can be solved by applying algorithmic thinking, pattern recognition, and logical reasoning.	
Procedural *What should they be able to do?*	Skills: • Review and apply the four pillars of computational thinking (decomposition, pattern recognition, abstraction, algorithm design). • Design and analyse algorithms for navigating mazes. • Compare the efficiency of different algorithms. • Use flowcharts to represent and understand algorithms.	• Implement decision-making (if/then/else) in algorithms. • Create algorithms to solve logic puzzles. • Understand and apply search algorithms. • Create and use simple ciphers to encrypt and decrypt messages. • Apply computational thinking skills in a collaborative escape room-style activity. • Use computational tools to aid in problem-solving.	
Disciplinary Literacy *(Tier 3 Vocab)*	• Computational Thinking • Algorithm • Flowchart • Pseudocode • Decomposition	• Pattern Recognition • Abstraction • Efficiency • Search Algorithm	• Logic Puzzle • Cipher • Encryption • Decryption
Assessment	Formative Assessment: • **Do Now Recaps**: Daily "Do Now" activities will assess prior knowledge and understanding from previous lessons. • **End of Lesson MS Forms**: Formative quizzes administered through Microsoft Forms will gauge student comprehension of concepts covered during the lesson. • **Designing and navigating algorithms:** Students design and implement a maze-solving algorithm using a flowchart tool. Summative Assessment: • **Digital Portfolio:** Review of the student's documented progress, demonstrating understanding of concepts and application of skills throughout game development. • **Unit Assessment:** Computational Thinking		

Curriculum Knowledge Map 2024-25

Year 8

Unit 3: Programming Puzzles: Building a Quiz with Python

Rationale:

Building on foundational understanding of the binary system and hardware, this unit aims to empower students to harness the power of computational thinking to solve problems and create interactive experiences. By exploring the creation of a simple quiz game, students will deepen their understanding of core programming concepts such as sequence, selection, and iteration, and how they relate to algorithms. This unit will not only bridge the gap between theoretical knowledge and practical application but also spark students' creativity and interest in programming, setting the stage for more advanced projects and exploration in the future.

Declarative

What should they know?

Definitions:

- **Pseudocode**: A simple, human-readable way to describe the steps of an algorithm using plain language and keywords.
- **Python**: A high-level programming language known for its readability and versatility.
- **Syntax**: The set of rules that define the structure of statements in a programming language.
- **Input**: Data that is provided to a program from an external source (e.g., user input).
- **Output**: Information or results that a program displays or sends to an external destination.
- **Variable**: A named storage location in a program that can hold a value.
- **Data Type**: A classification of data that determines the possible values it can hold and the operations that can be performed on it.
- **Casting**: Converting a value from one data type to another.
- **Processing**: The manipulation or transformation of data by a computer program.
- **Sequence**: The order in which instructions are executed in a program.
- **Selection** (Conditional Statements): The ability of a program to make decisions based on conditions (if, elif, else).
- **Iteration** (Loops): The repetition of a block of code multiple times.
- **Debugging**: The process of finding and fixing errors in a program.

Facts:

- Flowcharts can be used to visually represent the steps of an algorithm.
- Pseudocode helps bridge the gap between an algorithm and actual code.
- Python uses indentation to define blocks of code.
- Common errors in programming include typos, incorrect indentation, and logical mistakes.

Procedural

What should they be able to do?

Skills:

- Analyse the requirements of a quiz game and break down its functionality into smaller tasks (decomposition).
- Identify patterns in the structure and flow of quiz games (pattern recognition).
- Focus on the essential elements of a quiz game (abstraction) and design a step-by-step plan (algorithm) for its implementation.
- Write pseudocode to represent the logic of the quiz game.
- Understand basic Python syntax and use print() and input() for input and output operations.
- Declare and use variables of different data types (strings, integers, Booleans).
- Convert data types using casting when necessary.
- Write Python code that executes sequentially.
- Use if/else statements to create conditional logic in the quiz game.
- Implement basic error handling in Python code.
- Use print statements to debug and trace errors in the code.
- Explore using loops (for or while) to repeat questions or the entire quiz.

Disciplinary Literacy

(Tier 3 Vocab)

- Pseudocode
- Python
- Syntax
- Input
- Output

- Variable
- String
- Integer
- Boolean
- Casting (optional)

- Sequence
- Selection (Conditional Statement)
- If/Else Statement
- Iteration (Loop)
- Debugging

Assessment

Formative Assessment:

- **Do Now Recaps**: Daily "Do Now" activities will assess prior knowledge and understanding from previous lessons.
- **End of Lesson MS Forms**: Formative quizzes administered through Microsoft Forms will gauge student comprehension of concepts covered during the lesson.
- **Pseudocode/Flowchart Checks:** Pseudocode and flowcharts analysed for clarity, accuracy, and adherence to the quiz game's logic.

Summative Assessment:

- **Quiz Game Project: The final quiz game project will serve as the primary summative assessment.**
- **Unit Assessment:** Python Fundamentals