The New Senior Secondary Curriculum for Sierra Leone

Subject Syllabus for Computer Mathematics
Subject stream: Mathematics and Numeracy

This subject syllabus is based on the National Curriculum Framework for Senior Secondary Education. It was prepared by national curriculum specialists and subject experts.

SSSC
Senior Secondary School Curriculum

Curriculum elements for Computer Mathematics – an applied subject

Subject Description

Computer Mathematics aims to provide students with the requisite knowledge to gain an understanding of the mathematical concepts that underpin computers and how they store, process, communicate and transmit data. The students will be able to link theoretical mathematical concepts with its practical applications in computer programming, from logical and expressions and control structures for program flow, to matrices and functions for sophisticated data capture and analysis.

Rationale for the Inclusion of Computer Mathematics in the Senior Secondary School Curriculum

Computer Mathematics is a unique subject in the Senior Secondary School curriculum. Not only does it go in greater depth in topics such as Base Number Arithmetic and Logic, which are usually only briefly covered in a typical Mathematics course, it introduces concepts such as Algorithms not typically found in a secondary school curriculum. Its inclusion broadens and deepens the scope of mathematics knowledge the student is exposed to and enriches the curriculum in turn. Students will develop knowledge and skills through studying Computer Mathematics which they can build on in their future studies and employment. Computer Mathematics can be studied along Applications of Computer Mathematics (Coding) and together they make an impressive package for acquiring 21st century skills.

General Learning Outcomes

At the end of the course, students will be able to:

- explain the concept of number systems including the Real Number System and Base Number System
- use data representation and number base arithmetic
- describe logic connectives and construct truth tables for logic gates
- understand and use algorithms through writing pseudocode and creating flowcharts
- use logical, arithmetic and relational expressions in pseudocode and flowcharts
- describe and use control structures in pseudocode and flowcharts
- solve linear equations, inequalities and formulas
- understand and use Set Theory
- understand and use probability
- describe character encoding systems
- solve simultaneous linear equations graphically, by substitution and elimination
- explain the fundamental principle of counting and use it to calculate probability
- understand permutations and combinations, and use them to calculate probability
- describe a function, its domain and range, inverse and composite functions
- understand and use matrices, including matrix operations and finding transpose and inverse of matrices
- use matrices to solve systems of linear equations

Structure of the Syllabus Over the Three Year Senior Secondary Cycle

SSS 1	SSS 2	SSS 3
NUMBER SYSTEMS • Number Systems Concepts • The Real Number System • Properties of Real Numbers • Base Number Systems • Conversions Between Number Bases **DATA REPRESENTATION AND BASE ARITHMETIC** • Computing Number Bases • Units of Information • Four operations on Binary Numbers • Addition and Subtraction of Octal Numbers • Addition and Subtraction of • Hexadecimal Numbers	**FURTHER BINARY ARITHMETIC** • Unsigned and Signed Binary Numbers • Complements of Binary Numbers • The Four Operations on Unsigned and Signed Binary Numbers **CHARACTER ENCODING SYSTEMS** • More on Hexadecimals • Character Sets • Character Encoding Systems **LOGIC II** • Tautologies and Contradictions • Conditional Statements • De Morgan's Laws • Laws of Boolean Algebra • XOR, NAND and NOR Logic Gates	**MATRICES** • Basic Matrices Concepts • Addition and Subtraction of Matrices • Scalar Multiplication • Matrix Multiplication • Properties of Matrix Operations • Determinant of Matrices • Matrix Row Operations • Inverse Matrices **SYSTEMS OF LINEAR EQUATIONS** • Solve Linear Equations in Two and Three Variables: Inverse Matrix Method • Solve Linear Equations in Two and Three Variables: Gaussian Elimination Method • Solve Linear Equations in *n* Variables: Algorithm
LOGIC I • Basic Logic Concepts • Statements and Logical Connectives • Truth Tables • Boolean Logic **ALGORITHMS** • Basic Algorithm Concepts • Pseudocode • Flowcharts	**SIMULTANEOUS LINEAR EQUATIONS** • Solve Linear Equations in Two Variables: Graphical Method • Solve Linear Equations in Two Variables: Elimination Method • Solve Linear Equations in Two Variables: Substitution Method • Solve Linear Equations in Two Variables: Word Problems • Solve Linear Equations in Three Variables: Graphical Method • Solve Linear Equations in Three Variables: Elimination Method • Solve Linear Equations in Three Variables: Word Problems	**REVISION** • All Topics

LINEAR EQUATIONS, LINEAR INEQUALITIES AND FORMULAS • Basic Algebra Concepts • Linear Equations in One Variable • Linear Inequalities in One Variable • Graphical Representation of Linear Inequalities • Formulas	**SET THEORY II** • Cartesian Products of Sets • Partition of Sets • Power Sets	
SET THEORY I • Basic Set Concepts • Types of Sets • Venn Diagrams • Subsets and Proper Subsets • Set Operations • Properties of Set Operations • De Morgan's Laws **PROBABILITY I** • Basic Probability Concepts • Experimental and Theoretical Probability • Probability of Events • Mutually Exclusive Events • Independent Events • De Morgan's Laws • Conditional Probability	**PERMUTATIONS, COMBINATIONS AND PROBABILITY** • Fundamental Principles of Counting • Multiplication Principle: Factorial Notation • Permutations • Combinations • Probability of Events **FUNCTIONS** • Mappings, Relations and Functions • Using Function Notation • Types of Functions • Representing Functions • Domain and Range of Functions • Inverse Functions • Composite Functions	

Structure of the Syllabus Over the Three Year Senior Secondary Cycle

	SSS 1	SSS 2	SSS 3
Term 1	**NUMBER SYSTEMS** • Number Systems Concepts • The Real Number System • Properties of Real Numbers • Base Number Systems • Conversions Between Number Bases **DATA REPRESENTATION AND BASE ARITHMETIC** • Computing Number Bases • Units of Information • Four operations on Binary Numbers • Addition and Subtraction of Octal Numbers • Addition and Subtraction of • Hexadecimal Numbers	**FURTHER BINARY ARITHMETIC** • Unsigned and Signed Binary Numbers • Complements of Binary Numbers • The Four Operations on Unsigned and Signed Binary Numbers **CHARACTER ENCODING SYSTEMS** • More on Hexadecimals • Character Sets • Character Encoding Systems **LOGIC II** • Tautologies and Contradictions • Conditional Statements • De Morgan's Laws • Laws of Boolean Algebra • XOR, NAND and NOR Logic Gates	**MATRICES** • Basic Matrices Concepts • Addition and Subtraction of Matrices • Scalar Multiplication • Matrix Multiplication • Properties of Matrix Operations • Determinant of Matrices • Matrix Row Operations • Inverse Matrices **SYSTEMS OF LINEAR EQUATIONS** • Solve Linear Equations in Two and Three Variables: Inverse Matrix Method • Solve Linear Equations in Two and Three Variables: Gaussian Elimination Method • Solve Linear Equations in *n* Variables: Algorithm
Term 2	**LOGIC I** • Basic Logic Concepts • Statements and Logical Connectives • Truth Tables • Boolean Logic **ALGORITHMS** • Basic Algorithm Concepts • Pseudocode • Flowcharts **LINEAR EQUATIONS, LINEAR INEQUALITIES AND FORMULAS** • Basic Algebra Concepts • Linear Equations in One Variable	**SIMULTANEOUS LINEAR EQUATIONS** • Solve Linear Equations in Two Variables: Graphical Method • Solve Linear Equations in Two Variables: Elimination Method • Solve Linear Equations in Two Variables: Substitution Method • Solve Linear Equations in Two Variables: Word Problems • Solve Linear Equations in Three Variables: Graphical Method • Solve Linear Equations in Three Variables: Elimination Method • Solve Linear Equations in Three Variables: Word Problems	**REVISION** • All Topics

	• Linear Inequalities in One Variable • Graphical Representation of Linear Inequalities • Formulas	**SET THEORY II** • Cartesian Products of Sets • Partition of Sets • Power Sets	
Term 3	**SET THEORY I** • Basic Set Concepts • Types of Sets • Venn Diagrams • Subsets and Proper Subsets • Set Operations • Properties of Set Operations • De Morgan's Laws **PROBABILITY I** • Basic Probability Concepts • Experimental and Theoretical Probability • Probability of Events • Mutually Exclusive Events • Independent Events • De Morgan's Laws • Conditional Probability	**PERMUTATIONS, COMBINATIONS AND PROBABILITY** • Fundamental Principles of Counting • Multiplication Principle: Factorial Notation • Permutations • Combinations • Probability of Events **FUNCTIONS** • Mappings, Relations and Functions • Using Function Notation • Types of Functions • Representing Functions • Domain and Range of Functions • Inverse Functions • Composite Functions	

Teaching Syllabus

Topic/Theme/Unit	Expected learning outcomes	Recommended teaching methods	Suggested resources	Assessment of learning outcomes
YEAR 1/TERM 1				
NUMBER SYSTEMS Number Systems Concepts The Real Number System Properties of Real Numbers Base Number Systems Conversions Between Number Bases	Students will be able to: Show they understand the concept of number systems Show they understand the structure of the Real Number System and the relationship between the numbers Classify a real number as natural, whole, integer, rational or irrational Order and compare real numbers Show they understand and can apply properties of addition and multiplication of real numbers, i.e., for two real numbers a and b, including: ◦ closure ($a + b \in \mathbb{R}$ and $ab \in \mathbb{R}$) ◦ commutative ($a + b = b + a$ and $ab = ba$) ◦ associative ($(a+b) + c = a + (b+c)$	Introduce number systems as systems used to express or represent numbers Guide students to work in pairs to investigate different types of number systems using the provided information sheets Demonstrate and guide students to describe the set of real numbers. Guide students to use basic set notation and Venn diagrams to show the inter-relationship between the numbers Guide students to work in pairs to sort a set of 20 real number cards according to the type of number	Textbook Information sheets on different types of number systems: ◦ additive (e.g., Roman) ◦ multiplicative (e.g., Chinese) ◦ cipher (e.g., Greek) ◦ positional (e.g., Hindu-Arabic, also known as the decimal) number systems Activity sheets Computer Internet Card sets of real numbers such as -3, $\frac{4}{7}$, $\sqrt{5}$, 12, etc., Labels for 'natural number', 'whole number', 'integers', 'rational number' and 'irrational number' Dot card sets for decimal, binary, octal and hexadecimal (i.e. dot patterns on one side, the	Students are able to: Investigate and take part in a class presentation on different types of number systems Describe the structure of real numbers using set notation and Venn diagrams Hold up one of the real number labels in response to a number written on the board or called out by the teacher Identify which subset of real numbers to put given numbers Put the correct inequality symbol between two real numbers List all the numbers in a given range larger (or smaller) than a given number Explain to a classmate using examples why $(a+b) + c = a + (b+c)$

	and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$) ◦ identity (or neutral) elements ($a + 0 = a$ and $a \cdot 1 = a$) ◦ inverse elements ($a - a = a \times \frac{1}{a} = 1$ for $a \neq 0$) Show they understand that subtraction and division are neither associative nor commutative Show they understand and can use the positional number systems of bases: decimal, binary, octal, hexadecimal, and their notations Convert between decimal, binary, octal and hexadecimal systems	Demonstrate and guide students to use the inequality symbols, <, ≤, >, ≥ to compare the numbers Demonstrate each of the properties using a combination of real numbers, i.e., integers, rational and irrational numbers, e.g., for the commutative property for multiplication, show that: $7 \times 8 = 8 \times 7$ Guide students to test some of the properties themselves, e.g., How can we show that $\frac{12}{\sqrt{14}} \times \frac{1}{\frac{12}{\sqrt{14}}} = 1$ For commutativity, guide students to show on a number line why they get the same answer for addition and multiplication of real numbers Guide students to compare what they get when they do the calculation, $6 - 2$ and $2 - 6$	number in the corresponding base on the other) Place-value charts for decimal, binary, octal, hexadecimal number systems	and $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ Answer questions such as: ◦ "Why is subtraction not commutative?" ◦ "Why is multiplication associative?" Show on a number line that real numbers are commutative and associative under: $9 + 4 = 4 + 9$ and $3 \times 9 = 9 \times 3$ Answer standard questions on properties of real numbers Answer questions such as: "What are the place values for base 2 (8 / 16)" Read and write a decimal, binary, octal or hexadecimal number using the correct base notation Create a table (or poster) of the first 16 decimal, binary, octal and hexadecimal numbers

		Do they get the same answer? Can they show the calculation on a number line to explain why they get different answers? Guide students to do the same for division and explain their result Demonstrate and guide student to show the positional (place-value) nature of these systems using place-value charts with the positions being represented by the powers of the base Guide students to use the base notation, e.g., 56_8 (for an octal number, or # or 0x prefix for hexadecimal Guide students to use dot cards to show how counting is done in decimal (base 10), binary (base 2), octal (base 8) and hexadecimal (base 16) Discuss correct counting, e.g., "56" in octal is not pronounced "fifty-six", but "five six"; fifty six is a		Work in pairs to match cards with the same decimal, binary, octal and hexadecimal number Work independently or in pairs to use the appropriate powers to convert between different base systems

		decimal number (five tens and six ones) Demonstrate using the appropriate powers of the base how to convert: • a number in decimal to binary, octal or hexadecimal, and vice versa • a number in binary to octal and vice versa • a number in binary to hexadecimal and vice versa		
DATA REPRESENTATION AND BASE ARITHMETIC • Computing Number Bases • Units of Information • Four operations on Binary Numbers • Addition and Subtraction of Octal Numbers • Addition and Subtraction of Hexadecimal Numbers	Students will be able to: Show an understanding of the context in which the binary number system is used in computer programming Show they understand how and why octal and hexadecimal number systems are used in computer programming Perform addition and subtraction on binary integer numbers Perform addition and subtraction on octal and hexadecimal integer numbers	Explain what bits and bytes are in terms of binary numbers Guide students to use the information sheets to show how binary numbers are used in computer systems. For example, a bit is a binary digit, 0 (for off) or 1 (for on). It is the unit of information computers use to store data (i.e., numbers, text, sound, images, etc.) Demonstrate and guide students to show how octal and hexadecimal numbers are used in writing code as they have fewer number of digits	Textbook Information sheets on the use of non-decimal number systems in computers Activity sheets	Students are able to: Explain to a friend who was absent why binary, octal and hexadecimal numbers are used in computer programming Can use conversions of decimal, binary, octal and hexadecimal numbers to explain data representation in computers Convert between units of information (e.g. bytes, kilobytes, megabytes, etc.) Use the place value of binary numbers to add and subtract two binary integer numbers

		which are easier to manipulate than binary numbers. For example: $89_{10} = 1011001_2$ $= 01011001_2$ $= 001\ 011\ 001_2$ $= 131_8$ Also, $89_{10} = 0101\ 1001_2$ $= 59_{16}$ Ask students how they would translate these numbers into bits Introduce the terms: byte (a group of 8 bits), nibble for half a byte (4 bits), kilobyte (KB), megabyte (MB), gigabyte (GB) and tetrabyte (TB) Remind students that computer storage (hard drive and RAM) is measured in (tens of) thousands of bytes Demonstrate how to complete the addition facts table for binary integer numbers, i.e. **+** **0** **1** **0** 0 1 **1** 1 10 11 111 + 101 1100		Use the place value of octal and hexadecimal numbers to add and subtract integer numbers

Use the addition table to guide students in adding two binary integer numbers, of no more than 8 bits per number, as shown above

Demonstrate and guide students to subtract two binary integer numbers of no more than 8 bits per number

Briefly explain overflow errors in calculations

For each base, guide students to complete the addition facts in tables e.g., Octal addition

+	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

Ask questions such as "Is there a pattern to the results obtained from adding two numbers together which can help us to add two octal (hexadecimal) numbers?"

		Use the addition facts to demonstrate how addition is performed on octal integer numbers. $\begin{array}{r} {}^{111} \\ 437 \\ +376 \\ \hline 1035 \end{array}$ Use similar reasoning to add hexadecimal integer numbers and to subtract in each of the bases. Limit the numbers to no more than 8 bits equivalent per number		
YEAR 1/TERM 2				
LOGIC I Basic Logic Concepts Statements and Logical Connectives Truth Tables Boolean Logic	Students will be able to: Understand the concept of mathematical logic and its applications in computer programming Identify open and closed statements Assign a truth value to a statement Write the symbolic form of simple statements Classify compound statements as a negation, conjunction, disjunction, conditional or bi-conditional Use logical connectives to	Ask students what they understand by the word 'logic'. Allow them to use their dictionaries and share the meaning found with the class. Guide students to use the information sheets to explain what logic is (i.e., a science that studies the principles of correct reasoning). Ask: "Does it match the dictionary meaning?" Discuss why/why not Guide students to research and discuss why logic is important in computer programming Discuss open and closed	Textbook Dictionary Information sheets on mathematical logic Internet Activity sheets Logic puzzles and games	Students are able to: research and write a brief report on the history of mathematical logic and its applications in computer programming Work in pairs to identify open and closed statements and give reasons for choice Assign a truth value to a statement by identifying them as true or false Work in pairs to form statements for their partner to identify as true or false and give reasons for choice Write simple statements using the form:

	write compound statements Write a compound statement given in words in symbolic form, and vice versa Use truth tables to analyse the truth values of compound statements Determine if statements are logically equivalent. Show they understand the concept of Boolean logic and its applications in computer programming Show they understand and can use operators in expressions to control the flow in computer programming: ◦ Boolean (or logical) operators ◦ relational operators: ▪ < (less than) ▪ <= (less than or equal to) ▪ > (greater than) ▪ >= (greater than or equal to) ▪ = (equal to) ▪ ≠ (not equal to) ▪ arithmetic operators	statements using everyday examples such as: *Mangoes are fruits* *Mangoes are the best fruits* Guide students to identify simple open and closed statements and discuss reasons for choices Demonstrate and guide students to use simple statements which can only be answered by 'True' or 'False' to explain the concept of mathematical logic, e.g. *4 is an even number* *Sierra Leone is in West Africa* *All prime numbers are odd* *1 + 2 = 3* Which of these are true (T) / false (F)? How do they know the statement is true/false? Can they justify their answer? Show how letters can be used to denote simple statements, e.g., let *p* represent the simple statement: *"Mariama is a farmer".*		Let *p*: grass is green Let *q*: every even number is divisible by 4 Work independently or in pairs to identify compound statements Use letters and the logical connectives to translate compound statements written in words to symbols and vice versa Write simple statements in words for a partner to form compound statements in symbols to represent: (*p* □ *q*) □ *r* and *p* □ (*q* □ *r*) Discuss in their pairs whether the two statements mean the same thing. Discuss as a class. Work independently or in pairs to complete logic puzzles and other logic games Use the truth tables for ~*p*, *p* □ ***q***, *p* □ *q* to construct new truth tables, e.g. for *p* □ ~*q*, *p* □ *p* □ *q*, etc.

		Show that this statement can be written as: Let *p:* Mariama is a farmer Demonstrate and guide students to classify compound statement as a negation or a combination of simple statements using logical connectives: • negation (not, ~) • conjunction (and, □) • disjunction (or, □) • conditional (if then, □) • bi-conditional (if and only if, □) Demonstrate using plenty of examples of increasing complexity to illustrate each of the compound statements. e.g. (of negation) the statement: *"Musa is at home"* has as its negation *"Musa is **not** at home"* Show how this can be written as: Let *p*: Musa is at home ~*p*: Musa is not at home ~*p* is read as "not *p*" Demonstrate that negations of true statements is always		Answer questions such as: ◦ "Under what conditions will a given compound statement be true?" Devise a strategy to help a friend systematically construct truth tables and thus determine its truth value Work independently or in pairs to complete logic puzzles and other logic games Draw truth tables to determine if statements are logically equivalent Explain Boolean logic and how it is used in computer programming Construct logic gates and truth tables for NOT, AND and OR from both everyday life and mathematics Write expressions using both Boolean and relational operators

false and vice versa

Guide students to interpret and write compound statements including negations in words and symbolic form

Introduce truth tables using an example in words for negation:
Let p: Musa is at home
$\sim p$: Musa is not at home

p	$\sim p$
T	F
F	T

If p is true, $\sim p$ is false.
If p is false, $\sim p$ is true

Explain that a truth table is used to determine whether a compound statement is true or false

Demonstrate further examples using mathematical statements, e.g.

Let p: 2 is an even number
$\sim p$: 2 is not an even number

		Guide the students to work in pairs. They use statements written in words to construct truth tables for negation, conjunction, disjunctive and the conditional statements Demonstrate using examples the conditions under which two statements are logically equivalent. Guide students to draw truth tables and understand this happens if and only if their truth tables are the same Introduce and discuss Boolean logic as the logic used to control the flow of computer programs Guide students to construct truth tables to represent the Boolean logic gates: ◦ NOT ($\bar{\ }$) ◦ AND (+), ◦ OR (·) where the binary digits 1 and 0 indicate True (on) and False (off) respectively Guide students to write the Boolean expression		

		for each gate Discuss and guide students to show how the Boolean operators are used giving examples from everyday life such as with the statements: *A:* I will take a keke if it is very hot *B:* I am tired Demonstrate and guide students to use the typical relational operators in conjunction with boolean operators, e.g. *if Age < 16 OR Age > 65 then you pay a reduced price* will result in either true or false depending on the age of the person Demonstrate and guide students to use the order of operations for Boolean operators, I.e. (Brackets) – NOT – AND – OR		
ALGORITHMS Basic Algorithm Concepts Pseudocode	Students will be able to: Define and state the characteristics of algorithms Outline functions of algorithms	Guide students to work in pairs to put a set of sequencing cards for a task, (e.g., making tea or coffee), in order. Alternatively ask students to think and write down	Textbook Activity sheets Sequencing cards Poster of flowchart names, symbols and uses	Students are able to: Explain what an algorithm is, its characteristics and functions Write algorithms to solve

Flowcharts	Explain and use the components of pseudocode to write simple algorithms for solving given problems Define and state the characteristics of flowcharts Classify flowchart symbols and their uses Use control structures to translate more complex algorithms to pseudocode and flowchart ◦ sequence: ▪ a linear execution of statements ◦ selection (conditionals): ▪ if/then ▪ if/then/else ▪ case ▪ Boolean logic ◦ repetition (loops): ▪ while ▪ for ▪ do/while ▪ repeat/until Draw flowcharts for solving given problems	the activities they need to do to complete the task Ask a few students to put their solutions on the board. Compare the solutions, ask questions on students' thinking (see Assessment) and vote on the most efficient (or quickest) solutions Explain what an algorithm is using the task students just completed Provide explanations of characteristics and functions of algorithms Show examples of simple algorithms written in pseudocode. Use them to explain the basic pseudocode (and coding) components: • variable – unknown quantity with a name, a data type, and a value. • assignment – give a value to a variable • transfer – read an input, write an output • control – specifies which is the next step to be executed		◦ a simple everyday problem, e.g., buying a mobile phone ◦ a simple mathematical problems, e.g. finding the area of a rectangle given the lengths of its sides Answer questions such as: ◦ "What made you decide to do it that way?" ◦ "What can you do to make your algorithm more efficient?" ◦ "What is the same / different between your solution and one on the board?" Use the components of pseudocode to write simple algorithms for given problems Problems include: ◦ adding three numbers ◦ finding the average of two or three numbers ◦ calculating perimeter, areas and volumes of shapes, etc. Answer questions such as: ◦ "What is a flowchart?" Explain to an absent classmate the characteristics,

		Guide students to write algorithms using pseudocode, e.g., add two numbers Show an example of a simple flowchart (e.g., for adding two numbers) and use it to define and state the characteristics of flowcharts Demonstrate using a prepared poster the names and pictures of flowchart symbols and their uses in drawing flowcharts Demonstrate and guide students to use pseudocode and flowcharts to show how more complex algorithms and programs are written using the three control structures Discuss sequential control as the default means by which a program is executed as used in the basic algorithms to date Explain how selection control is used to execute one or more statements if a given condition is met		names, symbols and uses of flowcharts Use the symbols of flowcharts to write simple algorithms for the given problems above Trace the logic of given pseudocode and flowcharts which show more complex problems using control structures Sse pseudocode and flowcharts to write more complex algorithms to solve a given problem. Problems include: ◦ find the largest / smallest number among three numbers ◦ generate the 5 times tables from 1x to 12x ◦ output the count of all even numbers between a user defined range of numbers ◦ check whether a number is prime or not (composite) ◦ write error message when input number is not 5 or 6 ◦ etc. Compare their algorithms with other students and improve their own

		Guide students to research the internet to find out how iteration control works {repeats a statement a certain number of times, or while a condition is fulfilled) Guide students to work in pairs to write pseudocode and draw flowcharts for more complex mathematical calculations, e.g., finding the larger /smaller of two numbers Compare and discuss students' pseudocodes for efficiency		
LINEAR EQUATIONS, LINEAR INEQUALITIES AND FORMULAS Basic Algebra Concepts Linear Equations in One Variable Linear Inequalities in One Variable Graphical Representation of Linear Inequalities Formulas	Students will be able to: Recall the basic concepts of algebra: ◦ order of operations ◦ evaluating algebraic expressions ◦ collecting like terms ◦ factorisation ◦ expanding single and double brackets ◦ rearranging and evaluating formulas Solve linear equations in one variable	Review the basic algebra concepts using examples and exercises for students to complete. Include algebraic expressions with fractions. Demonstrate collecting like terms using a combination of like terms, e.g., x^2y xy^2, etc. Include negative coefficients Demonstrate and guide students to solve equations of the type	Textbooks Activity sheets	Students are able to: Recall and answer questions on basic algebraic concepts Solve linear equations in one variable Check solutions to linear equations Solve word problems involving linear equations in one variable Solve linear inequalities in one variable and show the

	Solve word problems involving linear equations in one variable Solve linear inequalities in one variable Represent the solution sets of linear inequalities graphically Solve word problems involving linear inequalities in one variable Substitute values into formulas Change the subject of formulas	ax + b = c using the addition and multiplication principles Demonstrate other types, of linear equations e.g., with the variable on both sides of the equation, e.g. ax + b = cx + d. Use examples which will require expanding brackets, collecting like terms and fractions Guide students to check their solutions by substituting into the original equation Guide students to solve word problems involving linear equations in one variable Demonstrate and guide students to solve a variety of linear inequalities, e.g. 0 ≤ ax + b ≤ c Guide students to represent the solution on a number line Guide students to solve word problems involving linear inequalities in one variable		solution on the number line Solve word problems involving linear inequalities in one variable Substitute values into variables and solve Rearrange the formula, substitute values into variables and solve

		Demonstrate how to solve formulas by substituting values into scientific and other formula, e.g.: $I = PRT$, $a^2 + b^2 = c^2$ Guide students to solve similar formulas Demonstrate how to change the subject of a formula, i.e. make a different variable the subject, then solve by substituting given values		
YEAR 1/TERM 3				
SET THEORY 1 Basic Set Concepts Types of Sets Venn Diagrams Subsets and Proper Subsets Set Operations Properties of Set Operations	Students will be able to: define a set as a collection of objects with defined attributes Explain the use of Set Theory in computer programming Determine if a set is well defined Use the notations for naming sets and elements or members of a set List the elements of a set Write sets using the	Introduce sets using collections of objects, shapes, numbers and names from around the classroom, school and community Discuss why Set Theory is important in computer programming Guide students to use the internet to research and write a brief history of Set Theory and how it has influenced logic on which a lot of Computer Science is based Demonstrate and guide	Textbooks Activity sheets Computers Internet Objects or pictures of objects 2D attribute shapes Sets of number cards with common attribute e.g. odd, even, prime, square Sets of cards with names e.g., places, capital cities, flowers, surnames starting	Students are able to: Describe what a set is to their peers Explain why Set Theory is important to computer programming Give true or false answers to indicate whether collections of objects and numbers are well-defined sets Answer questions such as: ◦ "What are the elements in your set?" ◦ "What elements are not in your set?" ◦ "How can we make the set well-defined?"

	description method, roster form, and set-builder notation Use Venn diagrams to represent sets Classify sets as either finite or infinite Find the cardinality of a set Explain the conditions under which two sets are equal, equivalent, both or neither Identify disjoint sets Identify the unit set Identify, and use the notation for, the empty (null) set Identify, and use the notation for, the universal set Identify, and use the notation for, the complement of a set Use the notation for subsets of a set Find all subsets and proper subsets of a set Find the union of two sets	students to understand the concept of well-defined sets using examples and non-examples Demonstrate and guide students to name sets using capital letters, and elements using the ∈ symbol, i.e.: $p \in A$, which is read as: "p is an element of A" Guide students to use the proper notation for an element which does not belong to a set, i.e.: $p \notin A$ "p is not an element of A" Guide students to give everyday examples of sets and list elements of the set. Demonstrate how to write sets using: • description of their common attributes, e.g., the set of all natural numbers • roster form using curly brackets and ellipsis where appropriate, e.g. N = { 1, 2, 3, … } • set-builder notation, e.g.	with same initial	Write a given set in the other two ways Write sets in one of the forms from a table of data or graph, e.g. given a table of population of districts in Sierra Leone, they can write sets of districts with population more (or less) than 100,000, etc. Work independently or in pairs to sort a variety of given sets into finite, infinite, unit and empty sets Sort pairs of sets into equal, equivalent, both or neither Answer questions such as: ◦ "How many ways can you find to describe the set ...?" ◦ "What elements are in the complement of the set ….? Draw a Venn diagram to represent the set Describe five possible universal sets of which Sierra Leone is one of the elements, e.g., set of countries in West Africa, set of diamond exporting countries, set of countries who have hosted the Dakar rally, etc.

	Find the intersection of two sets Find the difference of two sets Show they understand and can use properties of set operations: ◦ commutative ◦ associative ◦ distributive ◦ identity ◦ dominative ◦ complement ◦ De Morgan's Law	N = { x : x is a natural number } read as: "set N is the set of all elements x such that x is a natural number" Provide examples of different sets. Guide students to work in groups to investigate each of the statements below by listing elements and drawing Venn diagrams of different types of sets: • finite set (a set with limited number of elements/members) • infinite set (a set with unlimited number of elements/members) • cardinality of a set (the (number of elements, *n*, in a set *A, n(A)*) • equal sets (two sets, *A, B,* containing exactly the same elements, i.e. *A* = *B*) • equivalent sets (two sets *A, B* containing the same number of elements, *n(A)* = *n(B)*) • disjoint sets (two sets *A, B* containing no		Answer questions such as: ◦ "What sets can you make for a different element such as …?" ◦ "Name another element and the different sets in which it can be put" ◦ "Name all the subsets of the set …" ◦ "Which of the subsets are also proper subsets of the set …?" Answer standard questions using set notation and Venn diagrams on subsets and proper subsets of a given set Answer standard questions using set notation and Venn diagrams on union, intersection and difference, of up to three sets Answer standard questions using set notation and Venn diagrams on properties of up to three sets

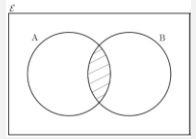
		elements in common) • unit set (a set, *A,* with a single member, $n(A) = 1$) • empty (null, Ø) set (a set with no elements or members) • universal set, *U or ε* (a set containing all the elements in a given context) • complement of a set (a set containing all the elements not in a particular set, A^c or A') Demonstrate and guide students to describe and draw Venn diagrams for: • subset (a set *A* containing all the elements of another set *B*, i.e. *A* □ *B*; sometimes the two sets can also be equal). Show that in a set with *n* elements the number of subsets is 2^n • proper subset (a set containing all the elements of another set, but is not equal to that set, i.e. *A* □ *B, A* □ *B*). Show that in a set with *n* elements the number of proper		

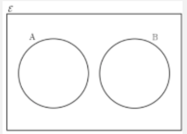
		subsets is $2^n - 1$ Discuss the following (or similar) example with the students: if A={2,4,6} then B={2,6} is a proper subset of A. The set C={2,4,6} is a subset of A, but it is not a proper subset of A since C=A. The set D={2,5} is not even a subset of A, since 5 is not an element of A. Demonstrate and guide students to use set notation and Venn diagrams to find: • union of two sets, $A \cup B$ • intersection of two sets, $A \cap B$ • difference of two sets, $A\|B$ (or A – B) Demonstrate and guide students to use set notation and Venn diagrams to show that given sets *A, B, C*: ◦ commutativity $A \cup B = B \cup A$ $A \cap B = B \cap A$ ◦ associativity $(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap$		

		C) ◦ distributivity *A* □ (*B* □ *C*) = (*A* □ *B*) □ (*A* □ *C*) *A* □ (*B* □ *C*) = (*A* □ *B*) □ (*A* □ *C*) ◦ identity *A* □ Ø = *A*, *A* □ *U* = *A* ◦ dominativity *A* □ *U* = *U*, *A* □ Ø = Ø ◦ idempotent *A* □ *A* = *A*, *A* □ *A* = *A* ◦ complement *A* □ A^c = *U*, *A* □ A^c = Ø ◦ De Morgan's Law (*A* □ *B*)^c = A^c □ B^c (*A* □ *B*)^c = A^c □ B^c Guide students to solve standard problems using these concepts *Demonstrate and guide students to look at other laws if time permits (e.g. double complement, absorption, etc.)*		
PROBABILITY I Basic Probability Concepts Experimental and Theoretical Probability Probability of Events Mutually Exclusive	Students will be able to: Demonstrate that they understand the language of probability Explain the use of probability in computer programming Find the experimental	Use practical examples, e.g., rolling a single die to explain the meaning of the words *experiment, event, outcome, sample space, fair, bias* Work independently or with a partner to classify given events according to whether they are *Certain*, *Likely*, *Unlikely* or	Textbooks Activity sheets Computers Internet Dice Playing cards Coins Spinners	Students are able to: Explain the terms used in probability, e.g., experiment, event, outcome, etc. Explain why probability is important to computer programming Use dice, playing cards,

Events Independent Events De Morgan's Laws Conditional Probability	probabilities of simple events Determine the sample space and theoretical probabilities for equally likely events Explain the difference between experimental and theoretical probabilities Show that the probability of any event, P, must satisfy $0 \le P \le 1$ Show that probabilities of all the events of an experiment add up to 1 Illustrate probabilities of simple events on a number line Understand and use the addition law for probabilities Describe mutually exclusive events and use the addition law to calculate probabilities Describe independent events and use the product law to calculate probabilities State and use De Morgan's Laws for probability Calculate simple conditional	*Impossible* Discuss why probability are important in computer programming Guide students to use the internet to research and discuss probability and how it is used in computer programming to understand the performance of algorithms, interpreting data, speech recognition (e.g. voice control phone access), etc. Guide students to perform simple experiments e.g. for drawing cards from a deck of playing cards, rolling a die, tossing a coin, spinning the pointer on a spinner etc. and record their results Demonstrate and guide students on how to find the experimental probability or relative frequency using the formula: $P(E)=\frac{\text{number of occurrences of an event}}{\text{total number of experiments}}$ $P(E)$ read as "probability of E" where E is the event		coins, spinners to conduct experiments and calculate the experimental probabilities of simple events Answer questions such as: ◦ "How many heads do you think you will get if you tossed a coin 5/10/50/100 times?" ◦ "What do you think the probability of getting a Queen of Hearts is? Try it and see if you are right." Write the sample spaces for the experiments above, and calculate the theoretical probabilities of simple events Explain to the class what the difference is between experimental and theoretical probabilities Answer questions such as: ◦ "Did you get the same answer for your experimental and theoretical probabilities?" ◦ "Why do you think your result came out the way it did?" ◦ etc. Calculate and verify that probabilities are between 0 and 1

	probability	in question Demonstrate and guide students to write the sample space for equally likely events and calculate their theoretical probabilities using the formula: $P(E)=\frac{\textit{number of outcomes of an event}}{\textit{total number of outcomes}}$ Guide students to use the results from their experiments and the theoretical sample spaces to differentiate between experimental and theoretical probabilities Discuss what happens with the experimental probability when a sufficiently large number of experiments is performed *Maybe move this up + the probability line* Demonstrate how individual probabilities of events are between 0 and 1, and all the probabilities add up to 1 Guide students to find		Calculate and verify that the probabilities of an experiment add up to 1 Show probabilities of events on a number line Use the addition law and Venn diagrams to calculate and show the probability of two events happening use the addition law for mutually exclusive events, and Venn diagrams to calculate and show the probabilities of mutually exclusive events Use tree diagrams to find the probability of events Use the product law to calculate the probabilities of independent events Use Venn diagrams to illustrate De Morgan's laws Solve problems using the principles of De Morgan's laws. Illustrate answers using Venn diagrams Answer standard questions on conditional probability

		probability of event *B*, given probability of event *A* as: $P(B) = 1 - P(A)$ Draw a probability line on the board and explain its features. Demonstrate and guide students to show the probabilities of simple events on the line Demonstrate using Venn diagrams the probability of two events happening  Show how this can be written as: $P(A) + P(B) - P(A \cap B)$ Guide students to use the addition law to find probabilities and show their results on Venn diagrams Demonstrate and illustrate using Venn diagrams to show mutually exclusive events as events which do not occur at the same time.		

		 Guide students to state and use the addition rule to find the probability of two mutually exclusive events *A* or *B* occurring, i.e., from the diagram: $(A \cap B) = 0 \Rightarrow P(A \cap B) = 0$ $\Rightarrow P(A \text{ or } B) = P(A) + P(B)$ Demonstrate using practical examples, e.g., the probability of getting a head and a tail at the same time when tossing a coin Introduce tree diagrams and use it to find simple probabilities Demonstrate and guide students to use tree diagrams to show two independent events defined as the probability of one event occurring having no effect on the probability of the other event occurring. Guide students to state and use the product rule		

		to find the probabilities of two independent events occurring, i.e. $P(A \text{ and } B) = P(A) \times P(B)$ Demonstrate using practical examples, e.g., the probability of getting a head on a coin toss and drawing a 3 of hearts from a pack of playing cards Discuss De Morgan's Laws (met previously in Sets) as it relates to probability, i.e. $P(A \cup B)^c = P(A^c \cap B^c)$ $P(A \cap B)^c = P(A^c \cup B^c)$ Demonstrate practical examples using Venn diagrams. Guide students to use the laws and Venn diagrams to solve problems Use Venn diagrams to illustrate and guide students to solve problems Guide students to calculate simple conditional probabilities i.e., Probability of event A, given that B had occurred is defined by:		

		$P(A\|B) = \frac{P(A \cap B)}{P(B)}$ Show using practical examples that conditional probability is the probability of event *A* occurring, given that event *B* occurs. Note if *A* and *B* are independent then $P(A\|B) = P(A)$ Guide students to use the formulas to calculate conditional probability of an event, e.g., given that a black card is drawn from a pack of cards, the probability of it being a seven, i.e., *P(seven|black)*		

YEAR 2/TERM 1

FURTHER BINARY ARITHMETIC Unsigned and Signed Binary Numbers Complements of Binary Numbers The Four Operations on Unsigned and Signed Binary Numbers	Students will be able to: Show that they know the difference between unsigned and signed binary numbers Show they understand and can use complements of binary numbers Recall and extend binary addition and subtraction Perform multiplication and division of unsigned binary integer numbers Perform multiplication and division of signed binary integer numbers	Review unsigned binary numbers Demonstrate and guide students to show signed numbers represented through • sign and magnitude notation (SM) using the most significant bit (MSB) • one's complement • two's complement Review how to add and subtract up to three unsigned binary numbers of no more than 8 bits per number Demonstrate and guide students to add and subtract unsigned and signed binary numbers using one's and two's complements Demonstrate and guide students to apply a binary shift to multiply and divide two unsigned binary integer numbers of no more than 8 bits per number Demonstrate and guide students to use two's	Textbooks Activity sheets	Students are able to: Differentiate between unsigned and signed binary numbers Explain and convert numbers using different representations Add and subtract unsigned and signed binary integer numbers using a variety of methods Multiply and divide unsigned and signed binary integer number using a variety of methods including binary shift and two's complement

		complement to multiply and divide unsigned and signed binary integer numbers of no more than 8 bits per number		
CHARACTER ENCODING SYSTEMS More on Hexadecimals Character Sets Character Encoding Systems	Students will be able to: Recall and extend understanding of hexadecimal numbers in computer programming Describe character sets in computer programming Understand and use character encoding systems: ◦ 7-bit ASCII ◦ Unicode	Recall that a number such as 89_{10} can be written as a 7-bit number, two groups of 4-bit numbers (nibble), and as 59_{16} as shown below: 89_{10} = 1011001_2 = $0101\ 1001_2$ = 59_{16} Demonstrate and guide students to write each hexadecimal (hex) from 0 to 255_{10} in groups of 4 bits Explain a character set as a defined list of characters (e.g., alphanumeric), recognised by the computer with each character (e.g. the letter q) represented by a number Demonstrate and guide students to use a character encoding table to: • convert characters to character codes	Textbooks Activity sheets Character encoding table for ASCII, Unicode	Students are able to: Write any decimal number from 0 to 255_{10} in hexadecimal Identify the types of characters found in a character set Convert characters to character codes and vice versa Write names of people and places in ASCII and Unicode Explain the benefits and limitations of ASCII and Unicode

		• convert character codes to characters Guide students to investigate the benefits and limitations of ASCII and Unicode		
LOGIC II Tautologies and contradictions Conditional Statements De Morgan's Laws Laws of Boolean Algebra XOR, NAND and NOR Logic Gates	Students will be able to: Identify tautologies and contradictions Recall and extend conditional statements to: ◦ identify the hypothesis and conclusion of a conditional statement ◦ convert statements to the standard ("if … then") form ◦ write the converse, inverse, and contrapositive of a conditional statement ◦ describe a counterexample for a conditional statement ◦ write the negation of a conditional statement. State and use De Morgan's laws to determine if statements are logically equivalent.	Discuss using examples; tautology, a statement which is always true; and contradiction, a statement which is always false. Guide students to identify these two types of statements from everyday and mathematical statements Use everyday and mathematical statements to demonstrate and guide students to extend their understanding of conditional statements Students use given statements and identify, write, convert and otherwise describe statements as required	Textbooks Activity sheets	Students are able to: Identify given statements as being a tautology or contradiction Use given statements and identify, write, convert and otherwise describe statements as required Use De Morgan's Laws to write statements that are equivalent to a given statement Draw truth tables for two statements to verify if they are logically equivalent Use the laws of Boolean algebra to solve standard logic questions including constructing truth tables Interpret the results of simple truth tables Solve problems on combinations of logic gates

	State and apply the laws of Boolean algebra ◦ commutative ◦ associative ◦ distributive Recall and extend use of the Boolean operators to include: ◦ XOR ◦ NAND ◦ NOR Describe more complex situations using combinations of logic gates	Recall equivalent statements as statements if and only if their truth tables are the same Discuss De Morgan's laws, (previously met in Sets in SSS1): • "not (A and B)" is logically equivalent to "not A or not B" $\sim(p \land q) \equiv \sim p \lor \sim q$ • "not (A or B)" is logically equivalent to "not A and not B" $\sim(p \lor q) \equiv \sim p \land \sim q$ Use examples to show the laws of Boolean algebra, including: ◦ commutativity $A \cdot B = B \cdot A$, $A + B = B + A$ ◦ associativity $(A \cdot B) \cdot C = A \cdot (B \cdot C)$ $(A + B) + C = A + (B + C)$ ◦ distributivity $A \cdot (B + C) = A \cdot B + A \cdot C$ $A + (B \cdot C) = (A + B) \cdot (A + C)$ These rules only apply to AND and OR Guide students to construct truth tables for these laws Review the NOT, AND and OR including		

		constructing truth tables for the logic gates Guide students to construct truth tables for the logic gates: ◦ XOR ◦ NAND ◦ NOR Guide students to write the Boolean expression for each gate Demonstrate and guide students to draw truth tables and write Boolean expressions for logic gates of increasing complexity		
YEAR 2/TERM 2				
SIMULTANEOUS LINEAR EQUATIONS Solve Linear Equations in Two Variables: Graphical Method Solve Linear Equations in Two Variables: Elimination Method Solve Linear Equations in Two Variables: Substitution Method Solve Linear Equations in Two Variables: Word	Students are able to: Solve linear equations in two variables using the graphical method Solve linear equations in two variables using elimination method Solve linear equations in two variables using the substitution method Solve word problems involving linear equations in two variables	Demonstrate and guide students to use the graphical method to solve two linear equation in two variables of the form: ax + by = c Use sets of equations of increasing complexity, e.g.: ◦ $5x + 2y$ □10 x □3 ◦ $5x + 2y$ □10 y □4 ◦ $5x + 2y$ □10	Textbooks Activity sheets	Students are able to: Solve linear equations in two variables using the graphical method Solve linear equations in two variables using elimination method Solve linear equations in two variables using substitution method Solve word problems involving linear equations in two variables using any

Problems Solve Linear Equations in Three Variables: Graphical Method Solve Linear Equations in Three Variables: Elimination Method Solve Linear Equations in Three Variables: Word Problems	Solve linear equations in three variables using the graphical method Solve linear equations in three variables using the elimination method Determine whether a given ordered triple is a solution to a three=by=three system of equations Solve word problems involving linear equations in three variables	$2x - y$ □4 Use the elimination and substitution methods to solve the same problems above Guide students to use any method to solve word problems involving two linear equations involving two variables Demonstrate and guide students to use the graphical method to solve three linear equation in three variables (known as three-by-three, 3×3, system) of the form: ax + by + cz = d Demonstrate how to solve three linear equations in three variables Demonstrate and guide students to substitute given triple into each equation in turn to verify whether or not it satisfies the equation. All three equations must be satisfied for the triple to be a solution Demonstrate how to set up and solve the three		method Solve linear equations in three variables using the graphical method Solve linear equations in three variables using the elimination method Substitute given triples in given three-by-three systems to verify if they are solutions Verify own solutions to check if they have a valid solution Solve word problems involving linear equations in three variables using the elimination method

		equations in three variables Guide students to set up and solve three-by-three systems of equations from word problem		
SET THEORY II Cartesian Products of Sets Partition of Sets Power Sets	Students will be able to: Describe and write the Cartesian product of *n* sets Calculate the size of the Cartesian product of sets Describe and find the partition of a set satisfying given conditions Describe and write the power set for small sets	Describe the Cartesian product of two sets as: A □ B = { (a, b) **:** a ∈ A, b ∈ A} i.e., "the set of ordered pairs (a, b) such that a ∈A, b ∈B" Demonstrate and guide students to write the Cartesian products of two sets, e.g. A = (a, b) and B = (1, 2), then A □ B = { (a, 1), (a, 2), (b, 1), (b, 2) } Ask students to write the Cartesian product for three given sets A, B, C. How would they generalise the product for *n* sets? Demonstrate and guide students to calculate the size (or cardinality) using the equation: |A □ B| = |A| □ |B| where |A|is the number of	Textbooks Activity sheets	Students are able to: Describe the Cartesian product of sets Write the Cartesian product of up to 3 sets Answer questions such as: ◦ "How do you know you have found all the sets?" ◦ "Is there a systematic way you can list so you have all the sets?" Equate elements of ordered pairs to find missing values, e.g. find x, y where: (2x, 1)= (3, y) Calculate the size of Cartesian product of sets Check by counting the elements of actual sets sets that the relation holds Partition a given set Given the partitions, of a set,

		elements in A Describe the partition of a set as a grouping of its elements into non-empty subsets, in such a way that every element is included in exactly one subset Guide students to partition sets in at least two different ways, e.g. S = { 1, 2, 3, 4, 5, 6, 7, 8 } One possible partition is: { 1 }, { 2, 3, 4 }, { 5, 6, 7, 8 } Another partition is: { 1, 2 }, { 3, 4 }, { 5, 6, 7, 8 } etc. Guide students to know and use the conditions under which a set is partitioned, e.g. there should be no null set Demonstrate and guide students to first find all the subsets of the given set, (we already know from SSS1, there are 2^n subsets) e.g. S = { a, b, c }, we expect 2^3 subsets, i.e. 8 subsets: { }, { a }, { b }, { c }, { a, b },		write the original set Write the power set of given sets Calculate the size of the power set of given sets

		{ a, c }, { b, c }, { a, b, c } Then guide students to write the power set P as: P = { { }, { a }, { b }, { c }, { a, b }, { a, c }, { b, c }, {a, b, c } }		
YEAR 2/TERM 3				
PERMUTATIONS, COMBINATIONS AND PROBABILITY Fundamental Principles of Counting Multiplication Principle: Factorial Notation Permutations Combinations Probability of Events	Students will be able to: Describe and apply the fundamental counting principles to solve simple problems Calculate the permutation of *n* objects using the multiplication principle Calculate the permutations of a set of *n* objects taken *r* at a time Define and calculate the number of combinations of a set of objects Calculate the probability of an event occurring	Describe the multiplication principle of counting: if there are *m* ways to do one task, and *n* ways to do another task, then there are $m \times n$ ways to do both tasks (this can be extended to doing 3 or more tasks) Ask pupils what this principle reminds them of (independent events in probability) Demonstrate and guide students to state how many ways, using this principle, there are of, for example, throwing a six on a die and tossing a head on a coin, then verify by completing the sample space Assist students to draw two-way tables and tree diagrams to help in enumerating all the outcomes from similar	Textbooks Activity sheets Calculators Computers Internet Dice Playing cards Coins Spinners	Students are able to: Identify what type of counting problem is in context Use appropriately the multiplication and addition principles of counting to solve simple counting problem Explain permutation to a peer Use the multiplication principle to calculate the permutation of *n* objects Calculate the permutations of a set of *n* objects taken r at a time Solve real-life problems on permutation Explain combination to a peer Calculate the number of combinations of a set of objects

		experiments and from everyday life Briefly introduce the addition principle of counting: if there are *m* ways to do one task, and *n* ways to do another task, and we cannot do both at the same time, then there are $m + n$ ways to do both tasks (this can also be extended to doing 3 or more tasks) Describe permutation as the arrangement of a number of objects, *n*, in order Demonstrate using the multiplication principle to count the ways of ordering the letters *P* and *Q* – *PQ* and *QP,* i.e., 2 or 2×1 ways Guide students to do the same for three letters, *P, Q, R* - there are 6 or 3×2×1 ways Assist students to arrange 4 letters – 4×3×2×1 or 24 ways Therefore, to arrange *n* letters (or objects) will		Solve real-life problems on permutation Calculate the probability of an event

		give $n!$ ways given by: $n! = n(n-1)(n-2)(n-3)...3\times2\times1$ read 'n factorial', with $0!=1$ Guide students to find the permutation of arranging *n* different objects taken *r* at a time, given by: ${}^{n}P_{r} = \frac{n!}{(n-r)!}$, $r \leq n$ (Assume no replacement) Demonstrate and guide students to use the appropriate button on their calculators to check their answers Discuss how this formula compares with the multiplication principle Describe combination as the selection of a number of objects in any order Guide students to find the combination of selecting *r* objects from *n* given objects, given by: ${}^{n}C_{r} = \frac{n!}{r!(n-r)!}$, $r \leq n$ (Assume no replacement) Review how to calculate the probability of an event from SSS1, i.e.		

		$P(E)=\frac{\text{number of outcomes of an event}}{\text{total number of outcomes}}$ Using examples, demonstrate and guide the students to calculate the required probability. Assist pupils in using a combination of the multiplication principle, the permutation formula and the combination formula, depending on the context of the question, to find the total number of outcomes and the number of outcomes of an event		
FUNCTIONS Mappings, Relations and Functions Using Function Notation Types of Functions Representing Functions Domain and Range of Functions Inverse Functions Composite Functions	Students will be able to: Describe mappings, relations and functions Understand and use function and function notation Explain the use of functions in computer programming Identify the different types of functions Represent functions ◦ using tables ◦ mapping diagrams ◦ graphically	Assist students to understand a mapping pairs each element of a given set (the domain) with one or more elements of a second set (the range). They are usually represented by moping or arrow diagrams Guide students to draw mapping diagrams to show the different types of mappings and how the elements are paired (e.g., one-to-one, onto, one-to-many, many-to-one, etc.)	Textbooks Activity sheets Graphic calculators Graph paper Computers Internet	Students are able to: Take part in a group discussion on mappings, relations, and functions and their inter-relationships Describe different types of mapping and their representations Draw mapping diagrams to represent relations and functions Explain what functions are and use the correct notations to describe them

	◦ algebraically ◦ as sets of ordered pairs Find the domain and range of a function Find inverse functions Find composite functions	Guide students to understand a relation as a collection of ordered pairs, (x, y), which are related by a rule and represented by a mapping diagram. Functions are relations which pairs one element in the domain with only one element in the range Describe the elements in the domain of a function as the set of independent variables, the input values, which the function processes; and the range as the set of dependent variables, the output values, which the function generates Guide students to use the notation x for the elements in the domain, and f(x) (read as 'function of *x*' or 'f of x'), as the function which generates the elements in the range. E.g. $f(x) = x^2$ read: '*f* of x equals x squared' Other letters are used which denote the same thing, e.g., g(x) ('g of x')		Research similarities and differences between mathematical and programming functions Make a presentation to their peers on the use of mathematical functions in computer programming Identify given functions as linear, quadratic, trigonometric, etc., or not a function Represent given functions in a variety of formats Find the domain and range of a given function Find the inverse of a given function Find the domain and range of the inverse of a given function find: *fog*, *gof*, $(fog)^{-1}$, $f^{-1}og^{-1}$, etc. Find the domain and range of the composite of a given function

		or *h*(x) Discuss the similarities and differences between mathematical functions and programming functions, (e.g., both types of function accepts input, does some processing and generates an output) Discuss why and how mathematical functions are used in computer programming. Discuss types of functions, e.g., • linear functions • quadratic functions • higher-order polynomial functions • rational functions • logarithmic functions • exponential functions • trigonometric functions • etc. Discuss how to identify a relation which is not a function, e.g., equation (graph) of a circle Describe each type of function and assist students to give the general formula and		

		examples of each type Demonstrate and guide students to represent functions in each of the ways given, including using graphic calculators and computers Provide a selection of functions in one format and guide students to represent them in one or more of the other formats, including as a set of ordered pairs Demonstrate and guide students to find the domain and range of functions given in a variety of formats Demonstrate and guide students to find the inverse, $f^{-1}(x)$ of a function, $f(x)$ given in a variety of formats, e.g. if $f(x)= 2x$, then $f^{-1}(x) = \frac{1}{2}x,\ x \in \mathbb{R}$ Guide students to make x the subject of more complex functions using $y = f(x)$ for convenience for $f(x) = 3x + 2$ rewrite $y = 3x + 2$ and make x the subject of the equation`		

		Discuss a composite function as a combination of two or more functions Guide students to find the composite of two functions $f(x)$ and $g(x)$ as: $fog(x) = f[g(x)]$ which is $g(x)$ followed by $f(x)$ and $gof(x) = g[f(x)]$ which is $f(x)$ followed by $g(x)$ Assist students to find the domain and range of the composite function Guide students to find the inverse of composite functions, as well as the composite of inverse functions		

YEAR 3/TERM 1

MATRICES Basic Matrices Concepts Addition and Subtraction of Matrices Scalar Multiplication Matrix Multiplication Properties of Matrix Operations Determinant of Matrices Matrix Row Operations Inverse Matrices	Students will be able to: Identify a matrix, stating its order and type Explain the use of matrices in computer programming Identify equal matrices and find missing elements Find the transpose of a matrix Find the sum and difference of two matrices Find scalar multiples of a matrix Find the product of two matrices Understand and use the properties of matrix operations: ◦ commutative ◦ associative ◦ distributive ◦ additive and multiplicative identities ◦ additive and multiplicative inverses Find the determinant of a matrix	Use examples of matrices to guide students to identify a matrix as a rectangular arrangement (called array) of items, usually numbers, into rows and columns. Show the notation for matrices and how elements of a matrix are identified Demonstrate and guide students to show how matrices are written and how to determine the order or dimension (m x n) and the notation for an element, a_{ij} Give examples of each type a matrix, e.g., square, triangular, row, column, identity, zero, etc. Ask students to write matrices for their peers to give the order and type. Students can also, give order and type for their peers to give examples Discuss why matrices are important in computer programming	• Textbooks • Activity sheets • Computers • Internet	Students are able to: Identify a matrix with m rows and n columns as of order m x n (m by n) Write a matrix using the correct notation Identify elements of a matrix using the correct notation, a_{ij} State the type of matrix depending on the number of rows, columns present Challenge their peers to identify the order and type of a variety of matrices Answer challenges from their peers to identify the order and type of a variety of matrices Explain why matrices are important to computer programming Identify equal matrices when given a variety of matrices Use equal matrices to find missing elements in a matrix Find the transpose of given

	Find the inverse of a matrix Perform row operations on a matrix	Guide students to use the internet to research and discuss matrices and how it is mainly used in computer programming to solve systems of linear equations, e.g., in graphics and image processing Use examples of matrices to demonstrate and guide students to identify equal matrices as having the same order, with the corresponding elements of the two matrices equal. Use that fact to guide students to find missing elements in either matrix Guide students how to find the transpose of an *m* x *n* matrix A by switching the rows and columns of the matrix, i.e., to get an *n* x *m* matrix, $(A^T)_{ij} = A_{ji}$ Demonstrate using examples how two matrices, A, B are added and subtracted provided they are of the same order: $(A + B)_{ij} = A_{ij} + B_{ij}$ and $(A – B)_{ij} = A_{ij} – B_{ij}$		matrices Check the order of matrices, then add or subtract as required Solve equations involving addition and subtraction of matrices Perform multiplication of a given matrix by a given scalar Give the condition for matrix multiplication to occur Find the product of two matrices Verify one or more of the properties using examples of matrices Investigate which matrix operations do not follow the properties and why Explain how to find the determinant of a 2×2 matrix Use their algorithm to find the determinant of a 2×2 matrix Extend the algorithm for finding determinants of 2×2 matrices to find the determinant of 3×3 matrices

		Guide students to add and subtract given matrices Demonstrate scalar multiplication to show how each element of the matrix is multiplied by the scalar to create another matrix of the same order and type $k(A_{ij}) = kA_{ij}$ Guide students to use this fact to carry out scalar multiplications Discuss using examples how scalar multiplication of two matrices A, B occurs only when the number of columns in the first matrix, i, is equal to the number of rows in the second, j, i.e. $(AB)_{ij} = a_i \cdot b_j$ Demonstrate and guide students to find the product of two matrices Show students the properties of operations of matrices using examples for *m* x *n* matrices A, B, C, e.g., for matrix addition: ◦ commutative		Find the inverse of a given 2×2 matrix using the standard algorithm Find the inverse of a given 3×3 matrix using the extended algorithm Perform and use the correct notation for single row operations on a matrix, e.g. $R_2 \rightarrow R_1$, $-3R_3 \rightarrow R_3$, $R_1+R_3 \rightarrow R_3$ Perform and use the correct notation for multiple row operations on a matrix, e.g. $2R_1 - 4R_3 \rightarrow R_3$

		A + B = B + A ◦ associative (A + B) + C = A + (B + C) ◦ additive identity A + 0 = 0 + A = A for a unique *m* x *n* matrix, 0 ◦ additive inverse A + (-A) = 0 = (-A) + A where -A is a unique *m* x *n* matrix Guide students to use matrices to prove these properties Demonstrate using a 2×2 matrix, A, the algorithm to find the determinant, symbolised by det(A) or |A|, of the matrix Guide the students to write their own algorithm using pseudocode and flowchart of how to find the determinant of a matrix. Guide them to exchange their algorithms with each other use to find determinants of 2×2 matrices Guide students to extend their algorithm to finding the determinant of 3×3 matrices		

		Share and discuss a few of the algorithms and make improvements Define the inverse, A^{-1}, of the matrix, A, as: $A \times A^{-1} = A^{-1} \times A = I$ where I is the identity matrix Demonstrate and guide students to find the inverse of a matrix using the determinant and the standard algorithm Guide students to extend the algorithm as demonstrated to find the determinant of 3×3 matrices Demonstrate and guide students to perform row operations on a matrix using the three operations: • switching rows • multiplying a row by a non-zero number • adding rows Demonstrate the notation for showing the row operation. For example: – switching rows 1 and 3 is written as R_1 □ R_3		

		– multiplying row 2 by 3 is written as $3R_2 \rightarrow R_2$ Guide students to perform these operations iteratively and in combination till they get the required matrix		
SYSTEMS OF LINEAR EQUATIONS Solve Linear Equations in Two and Three Variables: Inverse Matrix Method Solve Linear Equations in Two and Three Variables: Gaussian Elimination Method Solve Linear Equations in *n* Variables: Algorithm	Students will be able to: Explain why matrices are used to solve systems of linear equations Represent a linear system as a matrix Solve a 2×2 system of linear equations using an inverse matrix Solve a 3×3 system of linear equations using an inverse matrix Recognise the Gaussian elimination as an algorithm used to find the solution of a system of linear equations in *n* variables Write the augmented matrix of a system of equations Write the system of equations from an augmented matrix	Discuss why matrices are used to solve systems of linear equation Use a system of linear equations in two variables and demonstrate how it can be represented as a matrix by using the co-efficients of each equation to form a row of the matrix Guide students to make sure that both equations are in the linear equation form: $ax + by = c$ before they are written in matrix form: $Ax = B$ where A, B, x are matrices Review how to find determinants and inverse of a matrix.	Textbooks Activity sheets	Students are able to: Discuss why matrices are used to solve systems of linear equations Represent a 2×2 (or 3×3) linear system as a matrix Solve a system of 2×2 linear equations using an inverse matrix Solve a system of 3×3 linear equations using an inverse matrix Recognise Gaussian elimination method can be used to solve a system of linear equations Write an augmented matrix from a given linear system Write a linear system from a given augmented matrix Perform row operations on a matrix

	Perform row operations on a matrix Solve a 2×2 system of linear equations using Gaussian elimination Solve a 3×3 system of linear equations using Gaussian elimination Write an algorithm to solve a system of linear equations in *n* variables using Gaussian elimination	Using the matrix representation: Ax = B guide students to find the unknown variables as: $x = A^{-1}b$ Demonstrate and guide students to perform a matrix multiplication of A^{-1} and b, and equate the resulting matrix with the unknown variables Guide students to solve a system of 3×3 linear equations using an inverse matrix Discuss how the Gaussian elimination method uses matrices to solve systems. Show how the steps follow an algorithm (which can be written as a program) Demonstrate and guide students to write the system as an augmented matrix		Solve 2×2 systems of linear equations using Gaussian elimination Verify the solution to the linear system Solve and verify solution for 3×3 systems of linear equations using Gaussian elimination Use pseudocode and flowchart to write an algorithm to solve a system of linear equations in *n* variables

		Demonstrate and guide students to also be able to write a system from an augmented matrix Review how to perform row operations till they get the required matrix (an upper triangular matrix with all the element in the main diagonal equal to 1) Demonstrate and guide students to solve a 2×2 system of linear equations by following the algorithm: • represent the linear system as a matrix • write the augmented matrix of the system of equations • perform row operations on the matrix to get the required matrix • use back substitution to find the solution for each variable in the system Guide students to use Gaussian elimination to		

		solve a 3×3 linear system Guide students to write pseudocode and flowchart to solve a 2×2 system of linear equations using Gaussian elimination Extend the algorithm to an *n* by *n* system		

RESOURCES

Textbook
Information sheets on the use of non-decimal number systems in computers
Activity sheets
Computers / Smart Phones / Calculators
Internet
Objects or pictures of objects
2D attribute shapes
Sets of number cards with common attribute e.g., odd, even, prime, square
Sets of cards with names e.g., places, capital cities, flowers, surnames starting with same initial