SINGAPORE INTERNATIONAL MASTERY CONTESTS CENTER
SIMCC

National Junior
Informatics
Olympiad

Info Pack 2025

Internationl

National Junior Informatics Olympiad (NJIO)

The Value of SIMCC and STSF

National Contests in Countries Paper And Online

National Contests - Lowest Cost.

Country Awards :-
Top 8% - Gold,
Next 12% - Silver
Next 20% - Bronze
Next 10% - Honorable Mention

Global Contest Online only

Global Finals Online - Lower Cost.

Global Awards :
Top 10% - Gold,
Next 15% - Silver
Next 25% - Bronze
Next 20% - Honorable Mention

Global Finals (in person)

Global Finals - Travel Required, Reasonable Cost

Top 10% - Gold,
Next 15% - Silver
Next 25% - Bronze
Next 20% - Honorable Mention

SCIENCE	TECHNOLOGY	ENGINEERING	ARTS	MATHEMATICS
VANDA NATIONAL JUNIOR SCIENCE OLYMPIAD	National Junior Informatics Olympiad		SIAT Singapore International Arts Tournament	Singapore Math Challenge SMC; American Mathematics Olympiad; SASMO Singapore & Asian Schools Math Olympiad
		CS0 - Fundamentals of Computing; CS1 - Programming Methodology with C; CS2 - Data Structures and Algorithms		Mustang Math Tournament; SINGAPORE MATH GLOBAL FINALS
VANDA INTERNATIONAL JUNIOR SCIENCE OLYMPIAD	International Junior Informatics Olympiad	INTERNATIONAL	International Junior Arts Olympiad	SINGAPORE SIMOC; IJMO International Junior Math Olympiad

Scholarship Enhancement Initiatives

50 countries and territories

22 competitions and challenge

Over half a million contestants

01
Many internationally recognized competitions

SINGAPORE INTERNATIONAL MASTERY CONTESTS CENTER – SIMCC

02
SLSP, IJHS, YALA, GMOS Internships Passion with a purpose UGS

SLSP; YALA Young Achievers Leadership Academy; IJHS; GMOS Global Math Olympiad Scholarships

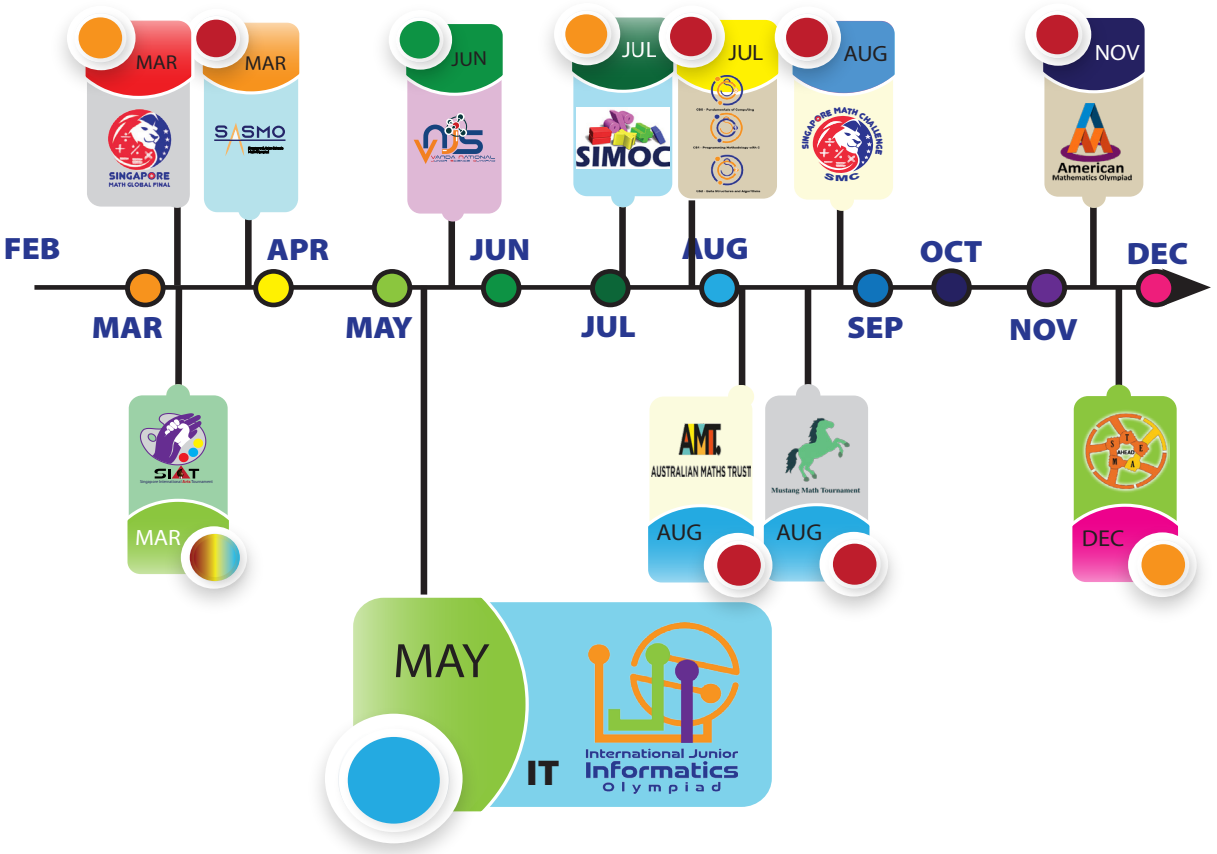
03
Best Fit University with Scholarship

International Academic & Cultural Contests STSF; University Global Scholars

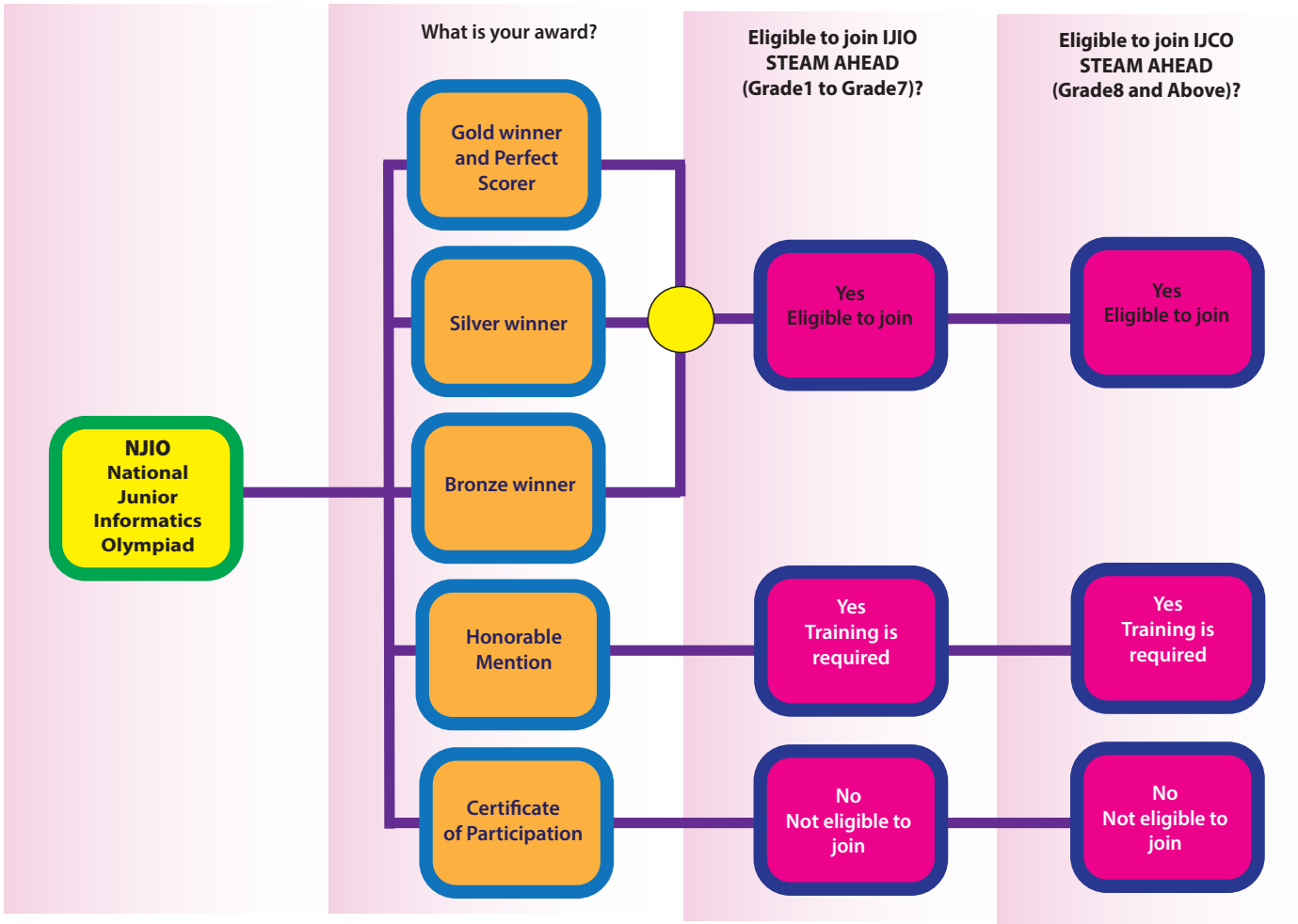
STEAM INTERNATIONAL TOURNAMENT

Over S$3 million in Scholarship Support to help students and teachers strengthen their portfolios and a Powerful Network to help our contestants to win more scholarships

SIMCC Competition Timeline



SIMCC Competition progression



STEAM INTERNATIONAL TOURNAMENT

There is a strong demand globally for STEAM talent led by corporations to get talent for their businesses. Hence, SIT is a great initiative to identify these talents for corporations and also various organisations giving scholarships to prepare talent for STEAM.

STEAM INTERNATIONAL TOURNAMENT (SIT)

SIMCC and Scholastic Trust Singapore (STS) is delighted to launch the STEAM International Tournament (SIT). SIT is a collection of reputable academic competitions in Science, Mathematics, Informatics, and Arts which help distinguish students' achievements in STEAM. Students who win awards in any of the qualifying contests below score points for the SIT Awards:

1. **Science** = Vanda National Junior Science Olympiad (VNJSO)
2. **Informatics** = National Junior Informatics Olympiad (NJIO) / Design thinking with robotics and Computational Thinking (DrCT) International Competition / National Junior Cybersecurity Olympiad(NJCO)
3. **Arts** = National Junior Arts Olympiad (NJAO) / Singapore International Art Tournament (SIAT)
4. **Mathematics** = American Mathematics Olympiad (AMO) and Singapore and Asian Schools Math Olympiad (SASMO)

Rules

1. Each student will be awarded SIMCC Scholarship points from each contest.
2. SIT points must be collected within one academic year from August 1 to July 31.
3. The SIT points collected will determine the SIT Award won for that year.

IJHS Scholarship points

Grant additional IJHS Scholarship points based on the combined awards received from AMO/SASMO, NJIO, NJOS, NJAO, and NJCO.

SIT Star Award ⟶ Earn extra 3 IJHS Scholarship points (SPs)

SIT Platinum Award ⟶ Earn extra 2 IJHS Scholarship points (SPs)

SIT Tri Award ⟶ Earn extra 1 IJHS Scholarship points (SPs)

SIT Award ⟶ Earn extra 0.5 IJHS Scholarship points (SPs)

SIT Star Awardees will be trained as SIMCC STEAM camp leaders in 2024 and awarded S$200 voucher upon completion.

Top 5 SASMO Winners from each grade by country* get Contest Scholarship (CS) to compete in MMT.

No Travel needed, and earn more IJHS Scholarship points to advance to top schools and universities with scholarships

Earn additional scholarship points

Table of scholarship points from all SIMCC competitions

SIT awards will be announced together with the induction of IJHS Scholars annually on September 15, 2024.

National Contest award	Perfect Score	Gold	Silver	Bronze
IJHS Scholarship Point	3	2	1	0.5

Combined Qualifying National Contest IJHS Scholarship Points	≥ 10 points	8 to 9 points	6 to 7 points	5 points
Award	SIT Star	SIT Platinum	SIT Tri	SIT Award
Extra IJHS Scholarship Point For Award	3	2	1	0.5

Contents

About NJIO

Overview

National Junior Informatics Olympiad (NJIO) an innovative approach designed to spark students' interest in programming by leveraging the principles of computational thinking and block programming. This statement explains how NJIO accomplishes this goal by breaking down complex problems into manageable parts, fostering an engaging and accessible learning environment.

Objectives

National Junior Informatics Olympiad (NJIO) integrates computational thinking into its core, a problem-solving process that involves understanding a problem, breaking it down into manageable parts, identifying patterns, abstracting the problem into a generalizable solution, and designing algorithms to solve it. This method teaches students to approach problems like a computer scientist, enhancing their analytical skills and making the programming process more logical and less intimidating. By engaging with computational thinking, students develop a foundational skill set that not only piques their interest in programming but also equips them with the tools necessary to tackle complex challenges in a systematic and efficient manner.

Making Programming Accessible with Block Programming: Block programming, a key component of NJIO, utilizes visual blocks that students can drag and drop to create programs. This approach eliminates the barrier of syntax errors, one of the most common hurdles for beginners in text-based programming. By focusing on the logic and structure of code rather than the intricacies of syntax, block programming makes programming more accessible and appealing to students, especially those who are just starting out or might be intimidated by traditional coding. . This hands-on, interactive method allows students to see immediate results of their logic, fostering a sense of accomplishment and motivating them to explore more complex concepts.

Interactive Learning Environment: NJIO offers an interactive and engaging learning environment that encourages experimentation. Students can see the real-time impact of their programming decisions, whether they're controlling a digital robot or creating an animation. This immediate feedback loop is crucial for learning and maintaining interest, as it allows students to experiment, learn from mistakes, and understand the principles of programming in a tangible way.

Encouraging Creativity and Innovation: By simplifying the programming process, NJIO allows students to focus on creativity and innovation. With computational thinking, they learn to see patterns and solutions in new ways, and with block programming, they can quickly implement and test their ideas. This freedom to experiment and innovate without the fear of failure is essential for fostering a deep interest in programming and technology.

NJIO, through its emphasis on computational thinking and block programming, demystifies the process of learning to code and makes it more accessible and enjoyable for students. By breaking down barriers to entry, encouraging problem-solving skills, and allowing for creative expression, NJIO not only sparks students' interest in programming but also prepares them with critical thinking skills that are applicable across various disciplines and in real-world scenarios.

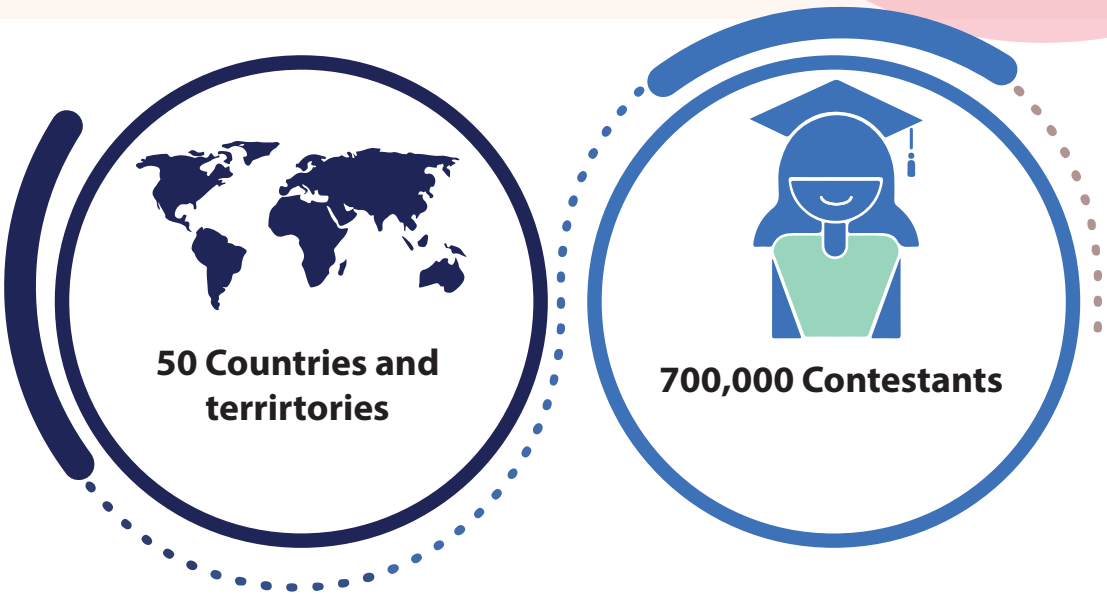
About STSF

Scholastic Trust Singapore Foundation (STSF) is a non-profit foundation that supports international academic and cultural competitions. With donations from revenue generated from these contests, STSF awards scholarships and mentorship programs for students and teachers to transform lives. Using knowledge building pedagogies and Singapore's expertise in education, STSF supports and mentors teachers and students through ACAs. We provide leading edge professional development for teachers in English, Mathematics, Science, IT, Pedagogy, and School Leadership to improve education in many developing countries. STSF manages the International Junior Honor Society (IJHS), Young Achievers Leadership Academy (YALA), a live-in 5-day 4 night leadership camp, and Southern Illinois University (SIU) Dr. Jared Dorn Scholarship and SIU International Student Tuition Grants.

About SIMCC

SIMCC is a social enterprise and donates 20% of her contest revenues to support students and teachers. SIMCC is one of the largest academic contest organizers in Singapore and Asia. We are committed to popularizing education through thinking games and competitions, and allowing students to interact, cooperate and build lasting bonds of friendship that transcend borders.

SIMCC has sales offices in Cambodia (Phnom Penh), Indonesia (Jakarta), and Singapore along with partners in 50 countries and territories with over half a million contestants in 22 competitions and assessments.



Benefits of NJIO

The NJIO contest offers a unique and innovative competition designed to enhance participants' computational thinking skills through engaging challenges that focus on block programming for primary school students. This contest provides significant benefits to both students and educators, sparking a deeper interest in programming and technology while fostering essential 21st-century skills.

For secondary school students, the focus shifts to textual programming languages, which are intensively used in computer science modules, particularly in topics like Abstraction and Algorithms, and Programming. Students will learn how to code solutions to real-world problems, developing computational thinking skills and applying the concepts and techniques taught. Additionally, they will develop systems thinking skills, particularly through the study of the Systems and Communications module.

SINGAPORE INTERNATIONAL MASTERY CONTESTS CENTER (SIMCC)

National Junior Informatics Olympiad (NJIO)
Lower Primary Division

Course/Contest(s):	NJIO
Grade(s):	1 & 2
Module/Contest Code:	ITG0102

Module Aims

Computational Thinking (CT) is a set of knwloedge and skills that involves breaking down complex problems into smaller manageable parts, and using logical reasoning and algorithms to solve them. This module aims to develop the critical thinking and problem-solving skills in students by imparting to them the introductory knowledge and skills of CT such as pattern recognition, sequencing, sorting, logical thinking and basic coding. Students will also learn how problems can be represented in text, images and graphs; and how they can be solved through optimization.

Contest Structure

Duration: 60 minutes

Section	Number of questions	Marks
Section A	10 CT questions	30 marks
Section B	5 Block programming questions	60 marks
Bonus points		10 marks
Total		100 marks

Notes:

Online

This is a CLOSE BOOK contest paper for National Junior Informatics Olympiad.

Section A:
3 marks for correct answer, 0 mark for unanswered questions, -1 mark for wrong answer

Section B:
12 marks for correct answer, 0 mark for unanswered or wrong answer

Reference

1. Dagiene, V., Sentance, S., & Stupuriene, G. (2017). Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics. Informatica (Netherlands), 28(1), 23-44. DOI: 10.15388/Informatica.2017.119.
2. Classic Computer Science Unplugged. Classic CS Unplugged
3. Computer Science Fundamentals. CS Curriculum for Grades K-5 | Code.org
4. Try Blockly. https://developers.google.com/blockly
5. Introduction to Microbit. https://makecode.microbit.org/courses/csintro/algorithms/overview
6. Bebras Challenge. https://www.bebraschallenge.org/
7. Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives: Complete edition, New York : Longman. Table 5.1, pages 67-68

Special Requirements

Prerequisite:	Nil
Software:	Coding tools like Blockly or Microbit
Hardware:	Windows 10 or Windows 11 PC / laptop, and / or other handheld smart devices.

Table of Specifications

	Abilities (%)			
Topics	**K**	**C**	**A/HA**	**Total**
A. Abstraction, Generalization and Decomposition	8	10	16	34
B. Data and Representations 1	6	8	7	21
C. Algorithms and Algorithmic Thinking 1	7	7	10	24
D. Basic Block Programming	6	6	9	21
Total	27	31	42	**100**

Notes:

1: The letters K, C, A and HA in the table of specifications denote the knowledge, comprehension, application and higher than application levels of Bloom's Taxonomy in the cognitive domain.

- Knowledge refers to the recalling of facts, concepts, procedures and theories.
- Comprehension refers to the ability to interpret, explain, infer, summarize, compare, classify and exemplify.
- Application refers to the ability to implement and / or execute.
- Higher than Application refers to the ability to analyze, evaluate, and create.

2: In the detailed syllabus which follows, all objectives should be understood to be prefixed by the words "At the end of instruction, the learner should be able to ...".

Content Outline

Topic A: Abstraction, Generalization and Decomposition

This topic introduces what Abstraction, Generalization and Decomposition are in the domain of computational thinking. These concepts are fundamentally important in critical thinking and problem solving. In Abstration, students will learn how unnecessary details are removed from a given problem and to focus on the important information. In Generalization, students will learn to identify patterns, similarities and connections. In Decomposition, students will learn to think about problems in terms of component parts and about how tasks can be broken into more managable sub tasks.

Topic B: Data and Representations

In this topic, students will be introduced to the Binary system of representing numbers. Binary system is the basic system that computers used to store information. Students will also learn how problems can be represented and interpreted in terms of text, images and graphs. This will enable students to better visualize and understand the problem.

Topic C: Algorithms and Algorithmic Thinking

In this topic, students will learn to use reasoning and organization to solve problems. In the process of reasoning and organizing, students will also learn to put instructions in the correct order or sequence or different ways the sequence can be arranged.

Topic D: Basic Block Programming

In this topic, students will be introduced to basic block programming using drag-and-drop tools. Online programming tools like Blockly and / or Microbit will be used, and students will learn how data like numbers, text, strings and instructions can be programmed and implemented in these platforms.

Objectives and Learning Outcomes

A.	Abstraction, Generalization and Decomposition
1	**Understand and Apply Abstraction**
	Objectives • Understand that abstraction means focusing on important details and ignoring unnecessary ones • Identify patterns and simplified representations of objects • Use symbols, pictures, and words to represent real-world things • Apply abstraction in problem-solving through sorting, grouping, and simplification. • Recognize how computers use abstraction **Learning Outcomes** • Understand **how abstraction helps simplify things** • Recognize **symbols, patterns, and simplified objects** • Organize information using **grouping and sorting** • Follow **simple instructions** and **write simplified steps** for common tasks • Relate abstraction to **technology, games, and coding**
2	**Understand and Apply Generalization**
	Objectives • Understand that **generalization means making a rule that applies to many things** • Identify **similarities and patterns** in objects, numbers, and words • Sort and group items based on **common characteristics** • Make **simple predictions** based on patterns • Recognize **general rules in real life** **Learning Outcomes** • Identify **similarities and patterns** in objects, numbers, and words • Make **simple predictions** based on what they have learned • Understand that **some rules apply** • Use generalization to **solve simple problems faster**

Objectives and Learning Outcomes

3	**Understand and Apply Decomposition**
	Objectives • Understand that **decomposition means breaking big problems into smaller steps** • Identify **parts of objects, stories, and tasks** • Organize tasks **step by step** for easier problem-solving • Use **sequencing and sorting** to make things clearer • Apply decomposition in **daily life, math, and simple programming** **Learning Outcomes** • Break down **big tasks into small steps** • Identify **parts of an object, story, or problem** • Arrange tasks in **the correct sequence** • Apply decomposition to **daily activities, math, and coding**
B.	Data and Representations
4	**Represent Data in Binary**
	Objectives • Introduce Binary number system • Understand how computers “think” using 0s and 1s • Develop basic pattern recognition skills • Relate binary numbers to **real-world applications** like computers and digital devices. **Learning Outcomes** • Recognize **0s and 1s as the building blocks of computers** • Count in **binary up to at least 5-bit numbers** • Understand **how binary is used in everyday technology** • **Translate simple words** into binary codes • Appreciate **computers & coding**!

Objectives and Learning Outcomes

5	**Represent Data in Text, Images and Graphs**
	Objectives • Understand that **data** is information that can be collected and represented • Identify how data can be stored and shared using **text, pictures, and graphs** • Read and create **simple pictographs, bar graphs, and tally charts** • Relate data to **real-world objects and everyday life** **Learning Outcomes** • Identify and classify **data in text, images, and graphs** • Understand **how pictures can tell stories** through data • Read and create **simple tally charts, pictographs, and bar graphs** • Relate data to **real-world experiences**
C	Algorithms and Algorithmic Thinking
6	**Use Logical Thinking and Sequencing**
	Objectives • Use **reasoning** to solve simple **real-life problems** • Identify **patterns and logical relationships** • Organize **objects, numbers, and ideas** in a structured way • Arrange **instructions in the correct order** to complete a task • Follow **rules and step-by-step directions** **Learning Outcomes** • Solve **simple problems using logical reasoning** • Sort and organize **objects, numbers, and ideas** • Follow **step-by-step instructions correctly** • Identify **errors in a sequence and fix them**

Objectives and Learning Outcomes

D	Basic Block Programming
7	**Understand Programming Tools**
	Objectives • Understand that **block programming is a way to tell computers what to do** • Learn **basic coding concepts** like sequences, loops, and events • Write simple programs using **Microsoft MakeCode for Micro:bit** • Control **LED lights, buttons, and sensors** on a Micro:bit • Apply coding skills in **real-world interactive projects** **Learning Outcomes** • Understand how **block programming works** • Write and modify **basic Micro:bit programs** • Use **loops, buttons, and sensors** to make interactive projects • Apply coding skills to **real-world applications**
8	**Implement Data in a Program**
	Objectives • Understand what **data** is and how it is used in a program. • Learn how to **store and use data** like numbers, words, and sensor inputs. • Recognize the importance of **correct sequencing in coding**. • Apply step-by-step logic to **write correct programs**. • Use **block programming tools like Micro:bit MakeCode or Scratch** to implement data and sequencing. **Learning Outcomes** • Understand how **data (numbers, text, inputs) is used in coding** • Implement **variables and basic decision-making** in block programming • Arrange **code in the correct sequence** for it to work properly • Apply **step-by-step debugging to fix errors** • Create simple **interactive programs using Micro:bit or Scratch**

SINGAPORE INTERNATIONAL MASTERY CONTESTS CENTER (SIMCC)

National Junior Informatics Olympiad (NJIO)
Middle Primary Division

Course/Contest(s):	NJIO
Grade(s):	3 & 4
Module/Contest Code:	ITG0304

Module Aims

This module aims to continue to build on Computational Thinking (CT) knowledge and skills for students in the mid primary stage. The module dwells deeper into CT thinking concepts and examples related to abstraction, decomposition, evaluation and generalization. The module also introduces how data can be represented in stack and memory and related mathematical concepts like sets, graphs, trees, permutations and combinations. Students will continue to learn algorithmic thinking skill by being introduced to well known algorithms releated to, for example, searching, sorting, shortest path, and optimization. Students will also enhance their computer knowledge, programming and logical thinking skills by learning security, and coding different kinds of variables, images, loops, conditionals to solve problems.

Contest Structure

Duration: 60 minutes

Section	Number of questions	Marks
Section A	10 CT questions	30 marks
Section B	10 Block programming questions	60 marks
Bonus points		10 marks
Total		100 marks

Notes:

Online
This is a CLOSE BOOK contest paper for National Junior Informatics Olympiad.

Section A:
3 marks for correct answer, 0 mark for unanswered questions, -1 mark for wrong answer

Section B:
6 marks for correct answer, 0 mark for unanswered or wrong answer

Reference

1. Dagiene, V., Sentance, S., & Stupuriene, G. (2017). Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics. Informatica (Netherlands), 28(1), 23-44. DOI: 10.15388/Informatica.2017.119.

2. Classic Computer Science Unplugged. Classic CS Unplugged

3. Computer Science Fundamentals. CS Curriculum for Grades K-5 | Code.org

4. Try Blockly. https://developers.google.com/blockly

5. Introduction to Microbit. https://makecode.microbit.org/courses/csintro/algorithms/overview

6. Bebras Challenge. https://www.bebraschallenge.org/

7. Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives: Complete edition, New York : Longman. Table 5.1, pages 67-68

Special Requirements

Prerequisite:	Nil
Software:	Coding tools like Blockly or Microbit
Hardware:	Windows 10 or Windows 11 PC / laptop, and / or other handheld smart devices.

Table of Specifications

	Abilities (%)			
Topics	**K**	**C**	**A/HA**	**Total**
A. CT Skills	5	6	8	19
B. Data and Representations 2	7	7	9	23
C. Algorithms and Algorithmic Thinking 2	8	8	15	31
D. Computers & Block Programming	7	7	13	27
Total	27	28	45	**100**

Notes:

1: The letters K, C, A and HA in the table of specifications denote the knowledge, comprehension, application and higher than application levels of Bloom's Taxonomy in the cognitive domain.
- Knowledge refers to the recalling of facts, concepts, procedures, and theories.
- Comprehension refers to the ability to interpret, explain, infer, summarize, compare, classify and exemplify.
- Application refers to the ability to implement and / or execute.
- Higher than Application refers to the ability to analyze, evaluate, and create.

2: In the detailed syllabus which follows, all objectives should be understood to be prefixed by the words "At the end of instruction, the learner should be able to ...".

Content Outline

Topic A: CT Skills

This topic continues to inculcate the Comptutational Thinking (CT) skills of Abstraction, Generalization, Decomposition and Evaluation. In Abstraction, students, given a problem, will learn how a representation of the system is chosen. In Generalization, students will learn to how new problems can be solved by adopting solutions from already-solved problems. In Decomposition and Evaluation, students will learn to think about problems in terms of sub-tasks or functions and how the best fitting solution can be obtained by optimizing resources based on certain constraints.

Topic B: Data and Representations 2

In this topic, students will be introduced to the concepts on data and information representation in Graphs, charts and tables as well as the scenarios where these can be applied. Students will also learn how data can be manipulated using Combinations, thus exposing them to their application in real-life examples of arranging and selection of resources and information.

Topic C: Algorithms and Algorithmic Thinking 2

In this topic, students will continue to build their algorithmic thinking skills by learning the key concepts of various algorithms and their real-life applications relating to the function of searching, sorting and finding the shortest path. Linear and Binary Searches will be used as examples.

Topic D: Computers & Block Programming

In this topic, students will learn the basics of how various loops and conditionals like if-then-else, repeat, do, and for loops are being programmed to solve problems.

Learning Outcomes

A.	CT Skills
1	**Apply Abstraction and Generalization**
	Objectives • Understand that abstraction means focusing on important details and ignoring unnecessary ones • Identify patterns and simplified representations of objects • Use symbols, pictures, and words to represent real-world things • Apply abstraction in problem-solving through sorting, grouping, and simplification • Recognize how computers use abstraction **Learning Outcomes** • Identify **what details are important and what can be ignored** (abstraction) • Find **patterns in numbers, science, and daily life** (generalization) • Simplify **complex tasks into a few easy steps** • Create **general rules** based on observations • Apply abstraction and generalization to **problem-solving, math, and coding**
2	**Apply Decomposition and Evaluation**
	Objectives • Understand that **decomposition means breaking a problem into smaller parts** • Apply decomposition to **solve problems step by step** • Learn that **evaluation means checking if a solution is correct and efficient** • Use **logical thinking and testing** to improve solutions • Apply decomposition and evaluation in **math, science, and coding** **Learning Outcomes** • Break big problems into **smaller, manageable steps** • Solve math and science problems using **decomposition** • Make better decisions by **evaluating different choices** • Identify and fix **mistakes in problem-solving and coding** • Apply decomposition and evaluation to **coding and daily life**

Learning Outcomes

B.	Data and Representations 2
4	**Data and Information representation**
	Objectives • Understand that **data is raw information** and must be organized to be useful • Learn different ways to **represent data** (text, numbers, pictures, graphs, tables) • Read and create **pictographs, bar charts, and simple digital representations** • Understand how computers **store and process data** using binary numbers • Apply data representation skills in **math, science, and simple programming** **Learning Outcomes** • Understand the difference between **data and information** • Read and create **pictographs, bar charts, and tables** • Recognize how computers **store and process data using binary numbers** • Apply data representation skills in **math, science, and coding**
5	**Apply Combination**
	Objectives • Understand that **combinations involve selecting items without considering order** • Learn the difference between **permutations (order matters) and combinations (order does not matter)** • Use **real-world examples** (e.g., choosing a fruit salad, selecting teams) • Apply **basic counting techniques** to find the number of possible combinations • Solve combination problems in **math, games, and coding** **Learning Outcomes** • Recognize when **order matters (permutation) vs. when it doesn't (combination)** • Use **tree diagrams and counting methods** to list combinations • Apply the **combination formula** to solve problems • Use combinations in **games, probability, and coding** • Understand the **importance of combinations in real-life decision-making**

Learning Outcomes

C	Algorithms and Algorithmic Thinking 2
6	**Use Search and Sort**
	Objectives • Understand the **importance of searching and sorting** in daily life and technology • Learn different types of **searching techniques** (Linear Search & Binary Search) • Explore **sorting methods** (Bubble Sort, Selection Sort, Insertion Sort) • Apply search and sorting skills in **math, real-world tasks, and coding** **Learning Outcomes** • Search for data using **Linear Search & Binary Search** • Sort numbers and objects using **Bubble Sort & Selection Sort** • Explain why sorting makes searching **faster and easier** • Apply searching and sorting in **daily life, math, and coding**
D	Computers & Block Programming
7	**Program Variables**
	Objectives • Understand that **variables store data** (numbers, words, or values) • Learn how variables **change during a program's execution** • Use variables in **math, real-life situations, and coding** • Apply variables in **Scratch, Micro:bit, or Python programs** **Learning Outcomes** • Understand what **variables are** and how they **store data** • Track and update **variables in real-life and coding scenarios** • Use **variables in Scratch, Micro:bit, and Python programs** • Apply variables in **math, games, and decision-making**

Learning Outcomes

8	**Program Loops and Conditionals**
	Objectives • Understand that **loops repeat actions** in coding • Learn that **conditionals (if-else) help programs make decisions** • Use loops and conditionals in **daily life, games, and coding** • Write programs in **Scratch, Micro:bit, or Python** using loops and conditionals **Learning Outcomes** • Understand **loops (repeat actions) and conditionals (make decisions)** • Apply loops and conditionals in **games, real-life, and coding** • Write simple programs in **Scratch or Python using loops and if-else** • Use loops to **repeat actions** and conditionals to **change behavior**

SINGAPORE INTERNATIONAL MASTERY CONTESTS CENTER (SIMCC)

National Junior Informatics Olympiad (NJIO)
Upper Primary Division

Course/Contest(s):	NJIO
Year/Stage(s):	5 & 6
Module/Contest Code:	ITG0506

Module Aims

This module for upper primary students builds upon the Computational Thinking (CT) knowledge and skills achieved in earlier stages. The module introduces functions and arrays commonly used in programming and their applications. Students will be refreshed on Searching and Sorting algorithms. In addition, Students will contiune to enhance their Data Structure and Representations knowledge in the areas related to Graphs, Finite State Machines and Trees. Students will also learn the basics of Cryptographic Protocols, Public-Key Cryptogrpahy and be introduced to floating point numbers, arrays, logical operations, nested loops and recursive functions and how they are programmed.

Contest Structure

Duration: 60 minutes

Section	Number of questions	Marks
Section A	10 CT questions	30 marks
Section B	10 Block programming questions	60 marks
Bonus points		10 marks
Total		100 marks

Notes:

Online
This is a CLOSE BOOK contest paper for National Junior Informatics Olympiad.

Section A:
3 marks for correct answer, 0 mark for unanswered questions, -1 mark for wrong answer

Section B:
6 marks for correct answer, 0 mark for unanswered or wrong answer

Reference

1. Dagiene, V., Sentance, S., & Stupuriene, G. (2017). Developing a Two-Dimensional Categorization System for Educational Tasks in Informatics. Informatica (Netherlands), 28(1), 23-44. DOI: 10.15388/Informatica.2017.119.

2. Classic Computer Science Unplugged. Classic CS Unplugged

3. Computer Science Fundamentals. CS Curriculum for Grades K-5 | Code.org

4. Try Blockly. https://developers.google.com/blockly

5. Introduction to Microbit. https://makecode.microbit.org/courses/csintro/algorithms/overview

6. Bebras Challenge. https://www.bebraschallenge.org/

7. Anderson, L. W., & Krathwohl, D. R. (Eds.). (2001). A taxonomy for learning, teaching and assessing: A revision of Bloom's Taxonomy of educational objectives: Complete edition, New York : Longman. Table 5.1, pages 67-68

Special Requirements

Prerequisite:	Nil
Software:	Coding tools like Blockly or Microbit
Hardware:	Windows 10 or Windows 11 PC / laptop, and / or other handheld smart devices.

Table of Specifications

	Abilities (%)			
Topics	**K**	**C**	**A/HA**	**Total**
A. Algorithms	7	10	10	27
B. Data Structures and Representations	8	7	7	22
C. Communications and Networking	6	7	6	19
D. Block Programming	9	8	15	32
Total	30	32	38	**100**

Notes:

1: The letters K, C, A and HA in the table of specifications denote the knowledge, comprehension, application and higher than application levels of Bloom's Taxonomy in the cognitive domain.
- Knowledge refers to the recalling of facts, concepts, procedures and theories.
- Comprehension refers to the ability to interpret, explain, infer, summarize, compare, classify and exemplify.
- Application refers to the ability to implement and / or execute.
- Higher than Application refers to the ability to analyze, evaluate, and create.

2: In the detailed syllabus which follows, all objectives should be understood to be prefixed by the words "At the end of instruction, the learner should be able to ...".

Content Outline

Topic A: Algorithms

In this topic, students will revisit the Linear Search and Binary search algorithms, the Bubble Sort, Insertion Sort and Selection Sort algorithm. Students will learn the approaches of breadth-first search and depth-first search. Another sorting algorithm, Merge Sort will be introduced to widen the students' knowledge on different sorting methods. Students will also be introduced to common algorithmic problems and how they are solved.

Topic B: Data Structures and Representations

In this topic, students will be introduced to the concepts on representing data and information in Arrays, Lists and Queues and the scenarios where these can be applied. In addition, students will learn what Finite State Machines (FSM) are and the way systems can be represented and modelled using Finite State Machines (FSM) to solve problems. More examples will also be introduced to illustrate the use of Trees.

Topic C: Communications and Networking

In this topic, students will enhance their knowledge of computer communications, network topology and security. Students will learn more about passwords, encryption, and cryptography like Public-Key Cryptography. Secured communications protocols like Secure Sockets Layer (SSL) and Transport Layer Security (TLS) will be introduced. Students will also learn the basics of network topology using different kinds of Graphs how problems relating to networks are solved using Graphs.

Topic D: Block Programming

This topic builds upon the foundation from the previous stages to enhance the programming skills of students. Students will learn how floating point variables are implemented, and how various loops like the while loop is coded. Students will learn how Boolean logic is implemented and how functions can be used to efficiently to solve more complex problems.

Learning Outcomes

A.	Algorithms
1	**Apply Advanced Search and Sort Algorithms**
	Objectives • Understand the **importance of searching and sorting** in daily life and computing • Learn different **searching techniques** (Linear Search & Binary Search) • Explore **sorting algorithms** (Bubble Sort, Selection Sort, Insertion Sort, Merge Sort) • Apply searching and sorting techniques in **math, real-life scenarios, and coding** • Write simple programs to implement **searching and sorting in Scratch & Python** **Learning Outcomes** • Explain **why searching & sorting are important** • Perform **Linear & Binary Search** manually and in code • Implement **Bubble Sort, Selection Sort, and Merge Sort** • Use **searching & sorting techniques in real-world and coding scenarios**
2	**Understand Well-Known Algorithmic Problems**
	Objectives • Understand what an **algorithm** is and why it is important • Solve well-known **algorithmic problems** such as sorting, searching, and pathfinding • Apply algorithmic thinking to **break down problems into smaller steps** • Implement basic algorithms in **Scratch, Python, or Micro:bit**. **Learning Outcomes** • Explain what an **algorithm is and why it's important** • Solve **common algorithmic problems using logical steps** • Use **searching & sorting algorithms efficiently** • Implement basic algorithms in **Scratch & Python** • Optimize solutions for **efficiency and speed**

Learning Outcomes

B.	Data Structures and Representations
4	**Represent Data in an Array, List and Queue**
	Objectives • Understand **arrays, lists, and queues** as data storage structures • Learn how **arrays store fixed-size data, lists grow dynamically, and queues follow FIFO (First-In, First-Out)** • Use **arrays, lists, and queues in math, real-life scenarios, and coding** • Write programs in **Scratch, Python, or Micro:bit** using these data structures **Learning Outcomes** • Understand **arrays, lists, and queues** and their differences • Use **arrays to store fixed data, lists to add/remove data, and queues to process data in order** • Implement these data structures in **Scratch, Python, and Micro:bit** • Apply data structures in **real-world applications, games, and problem-solving**
5	**Apply Finite State Machines and Trees**
	Objectives • Understand what a **Finite State Machine (FSM)** is and how it works • Learn how **Trees** store and structure information • Apply FSMs to **decision-making processes in games and real-life scenarios** • Use Trees to **organize data in hierarchy-based searches** • Implement FSMs and Trees in **Scratch, Python, or Micro:bit** **Learning Outcomes** • Understand how **Finite State Machines work in real-world and coding** • Use **decision trees and binary trees** to organize data • Implement FSMs and Trees in **Scratch & Python** • Apply FSMs & Trees in **games, AI, and real-life problem-solving**

Learning Outcomes

C	Communications and Networking
6	**Understand Secured Communications**
	Objectives • Understand why **secure communication is important** in daily life • Learn about **encryption, ciphers, and digital security** • Explore **common threats to communication (hacking, phishing, spying)** • Use **safe messaging practices and secure passwords** • Implement **basic encryption and decryption in Scratch or Python**. **Learning Outcomes** • Explain why **secure communication is important** • Identify **common threats to online security** • Use **basic encryption methods (Caesar Cipher, Substitution Cipher)** • Implement **secure messaging and passwords** in real life • Write programs in **Scratch & Python that encrypt and decrypt messages**
D	Block Programming
7	**Program Floating Point Variables and Loops**
	Objectives • Understand **floating-point numbers** and how they represent **decimal values** • Learn the difference between **integers and floating-point numbers** • Use **loops to repeat tasks** efficiently in programming • Write programs in **Scratch & Python that use floating-point variables and loops** **Learning Outcomes** • Understand the **difference between integers and floating-point numbers** • Use **loops to repeat tasks automatically** in programming • Modify **floating-point values using loops** • Write **basic programs in Scratch & Python that use loops and floating-point variables**

Learning Outcomes

8	**Program Boolean Logic and Functions**
	Objectives • Understand **Boolean logic (True/False, AND, OR, NOT)** • Learn how Boolean logic is used in **decision-making and conditions** • Create and use **functions to organize and reuse code** • Apply Boolean logic and functions in **Scratch, Python, and Micro:bit** **Learning Outcomes** • Understand how **Boolean logic (True/False) controls decisions** in programs • Use **Boolean operators (AND, OR, NOT) in coding** • Create and use **functions to organize and reuse code** • Write **basic programs in Scratch & Python using Boolean logic and functions**

National Junior Informatics Olympiad (NJIO)
Secondary Division

Course/Contest(s):	NJIO
Grade(s):	7 to 10
Module/Contest Code:	ITG0710

Module Aims

The core focus of the syllabus integrate both **ICT and Computer Science in the Curriculum.** The ICT and Computer Science components in the two syllabuses are designed to complement each other. In secondary school Computing, the ICT module leverages internet resources and spreadsheet application software to support students' understanding of computer science concepts.
Students will learn to apply selected mathematical, statistical, and financial functions, as well as conditional statements or expressions, within spreadsheet software to perform tasks. As a baseline ICT skill, they are expected to understand different data types, organize data into tables, create charts, and print data tables and visualizations.

The Computer Science modules place a strong emphasis on textual programming languages, particularly in **Abstraction and Algorithms** and **Programming**. Students will gain hands-on experience in coding solutions for real-world problems, developing computational thinking skills while applying learned concepts. Additionally, they will cultivate **systems thinking skills** through the study of the **Systems and Communications** module.

Core Areas of Study in Computing

The key concepts in computing, identified for syllabus design, are categorized into core areas of study. These areas and their definitions are:

- **Data and Information** – Methods for managing and processing data to transform it into meaningful and useful information for specific purposes.
- **Systems and Communications** – Approaches to integrating different components of a computer to function as a cohesive system.
- **Abstraction** – Techniques for analyzing and simplifying problems to facilitate problem-solving.
- **Algorithms** – Systematic methods for planning step-by-step solutions to problems.
- **Programming** – The process of instructing computers to execute solutions through code.

Curriculum Objectives

The curriculum objectives are designed based on key computing concepts and are structured across different educational levels—from upper primary to junior college—covering the five core areas with increasing depth. These objectives are implemented through syllabus aims, content, and learning outcomes to ensure a progressive learning experience for students.
At the upper secondary level, the curriculum objectives include:

1. Developing an understanding of algorithms that embody computational thinking.
2. Applying logical reasoning to evaluate the efficiency and effectiveness of algorithms and programming solutions in achieving desired outcomes.

3. Utilizing a textual programming language to solve diverse computational problems, incorporating appropriate data structures (e.g., lists, tables, arrays) and employing functions and procedures.
4. Understanding how data and instructions are represented and stored using binary digits (bits).
5. Gaining knowledge of computer network systems, including hardware and software components and their communication processes within and across systems.
6. Using computer technology safely and responsibly to develop meaningful products and services.

Contest Structure

Duration: 90 minutes

Section	Number of questions	Marks
Section A	7 CT questions	21 marks
Section B	12 programming questions	72 marks
Bonus points		7 marks
Total		100 marks

Notes:

Online
This is a CLOSE BOOK contest paper National Junior Informatics Olympiad.

Section A:
3 marks for correct answer, 0 mark for unanswered questions, -1 mark for wrong answer

Section B:
6 marks for correct answer, 0 mark for unanswered or wrong answer

Reference

2021 O-level Computing syllabus

Special Requirements

Prerequisite:	Recommended to complete the syllabus in Grade 5 & 6
Software:	Coding tools like Blockly or Microbit
Hardware:	Windows 10 or Windows 11 PC / laptop, and / or other handheld smart devices.

Table of Specifications

	Abilities (%)			
Topics	**K**	**C**	**A/HA**	**Total**
A. Data and information	6	10	12	28
B. System and communication	7	9	8	24
C. Abstraction and Algorithms	8	7	5	20
D. Programming	8	8	12	28
Total	29	34	37	**100**

Notes:

1: The letters K, C, A and HA in the table of specifications denote the knowledge, comprehension, application and higher than application levels of Bloom's Taxonomy in the cognitive domain.

- Knowledge refers to the recalling of facts, concepts, procedures and theories.
- Comprehension refers to the ability to interpret, explain, infer, summarize, compare, classify and exemplify.
- Application refers to the ability to implement and / or execute.
- Higher than Application refers to the ability to analyze, evaluate, and create.

2: In the detailed syllabus which follows, all objectives should be understood to be prefixed by the words "At the end of instruction, the learner should be able to ...".

Content Outline

Topic A: Data and Information

This topic focuses on data handling and processing within computer systems, emphasizing the importance of ethical considerations when working with data. Students will learn to identify various types of data, understand their purposes, and explain how data is structured or organized for processing and output, particularly in relation to specific problems. The module also aims to raise awareness of ethical concerns surrounding data, including privacy issues. It consists of two key units of study:

1.1 Data Management

1.2 Data Representation

Topic B: Systems and Communications

This topic covers systems that involve computer technology and computing devices. Students will explore the relationships between the components of a system and understand how each part contributes to enabling communication between humans and machines, between machines, and within a machine itself. The module consists of two key units of study:

2.1 Computer Architecture

2.2 Data Communications

Topic C: Abstraction and Algorithms

This topic focuses on problem-solving and the approach of breaking a problem down into smaller, more manageable parts to address each one individually. An algorithm provides a solution to the problem that is independent of any specific programming language and can be presented in either pseudo-code (where program structures are clearly defined) or diagrammatically (using flowcharts). Students will learn to distinguish between pseudo-code and flowcharts. The module consists of two key units of study:

3.1 Problem Analysis

3.2 Algorithm Design

Topic D: Programming

This module focuses on developing logical thinking, reasoning, and problem-solving skills through the design and creation of software solutions using programming languages. While an algorithm provides a language-independent solution, a programming language translates that solution into one that is executable on a computing device. Students will have the opportunity to test their algorithms by running the programming solutions to verify their effectiveness. The module consists of two key units of study:

4.1 Program Development

4.2 Program Testing

Learning Outcomes

A.	Data and Information
1	**Introduction to Data Management**

	Objectives
	
	This unit focuses on developing logical thinking and reasoning through data analysis, processing, and visual representation. Students will engage in hands-on activities using spreadsheet software to enhance their understanding of functions and formulas for computing and processing data.
	
	Learning Outcomes
	
	• Organize data by placing it under relevant column headings (e.g., data field names) and assigning appropriate data types (e.g., numeric, text, date).
	• Categorize and arrange data into columns with meaningful headings, explaining that these columns define the structure of the data table. Emphasize the importance of formatting columns correctly according to the data types used.
	• Utilize mathematical operators, functions, and what-if analysis (goal seeking) to create spreadsheets and solve real-world problems, such as: calculating the total, average, minimum/maximum, square root, simple interest, remainder of a division, rounding values, generating random values, converting data types (e.g., converting decimals to integers), and counting the number of data items.
	• Understand and apply conditional statements (both simple and nested), such as COUNTIF and IF, along with relational and Boolean operators like AND, NOT, and OR. This includes using conditional formatting.
	• Effectively use functions to look up data in rows or columns (both horizontal and vertical lookups) within lists or tables for data processing.
	• Common functions include:
	  ○ **Area Functions:** TODAY
	  ○ **Text Functions:** LEFT, LEN, MID, RIGHT
	  ○ **Logical Functions:** AND, IF, NOT, OR
	  ○ **Lookup Functions:** HLOOKUP, VLOOKUP
	  ○ **Mathematical Functions:** CEILING.MATH, FLOOR.MATH, MOD, POWER, QUOTIENT, RAND, RANDBETWEEN, ROUND, SQRT, SUM, SUMIF
	  ○ **Statistical Functions:** AVERAGE, COUNT, COUNTA, COUNTBLANK, COUNTIF, LARGE, MAX, MEDIAN, MIN, MODE.SNGL, SMALL

Learning Outcomes

2	**Data Representation**
	Objectives This unit explores how data is represented internally as binary numbers, the process of converting data between different number systems, and the applications of these number systems. The unit covers the binary, decimal, and hexadecimal number systems. **Learning Outcomes** • Represent positive whole numbers in binary form, where a bit (binary digit) is either 0 or 1. • Convert positive whole numbers between binary, decimal, and hexadecimal number systems, explaining the techniques used. The data should be represented using no more than 16 bits, and the conversion methods should be described concisely. • Describe the applications of number systems in contexts such as ASCII codes, IP (Internet Protocol) addresses, Media Access Control (MAC) addresses, and RGB color codes. • Explain how each number system is utilized in these specific areas.

B.	Systems and Communications
4	**Introduction to Computer Architecture**
	Objectives This unit focuses on the fundamental components of computer architecture and computer networks. Students will gain an understanding of the roles of hardware and software in a computer system or network, without needing to know the technical details of how they function. Through hands-on activities, students will work with hardware components and explore how hardware and software integrate and work together. **Learning Outcomes** • Describe the basic components of computer architecture, including the computer processor, memory, data and address buses, input (e.g., data and instructions), output (e.g., intermediate and final results of processing), and storage media. The focus is on understanding the "internals" of a computer without delving into technical details. • Identify logic gates based on their truth tables and evaluate Boolean expressions using truth tables. • Create truth tables for given logic circuits with up to three inputs. • Design and build simple logic circuits using AND, OR, NOT, NAND, and NOR gates.
5	**Introduction to Data Communications**

Objectives

This unit focuses on networks as a context for understanding resource sharing and data communication. Students will gain a general understanding of how data is transmitted, as well as the importance of ensuring data accuracy and security during transmission.

Learning Outcomes

- Identify and explain the function of various network hardware components, including modems, network interface controllers, hubs, switches, and routers.
- Describe the differences between wired and wireless networks, and explain the factors that influence the choice between the two.
- Name each device and explain its function, without needing technical details on how they operate. For example, it may be necessary to explain what a modem (modulator-demodulator) is. It is assumed that students are already familiar with the following input and output devices: monitor, keyboard, mouse, printer/plotter, scanner, and web camera.
- Describe the components of a simple home network and design one. Components may include IP (Internet Protocol) addresses, MAC (Media Access Control) addresses, ports, SSID (Service Set Identifier), and access points. Students may also participate in setting up a simple network.
- Compare and contrast client-server and peer-to-peer network models, focusing on:
    - Purpose
    - Function
    - Organization
    - Bandwidth (including speed)

      Function refers to how the network operates, while organization concerns the layout or connection of hardware components. No technical details about industry standards or materials used in network devices are required.

C	Abstraction and Algorithms
6	**Problem Analysis**

	Objectives
	
	This unit focuses on problem interpretation and analysis. Students will learn how to approach problem-solving in a structured way, using strategies such as breaking a problem into smaller, manageable parts and solving each part individually. They will have the opportunity to reinforce their understanding through hands-on activities, solving simple real-world problems.
	
	Learning Outcomes
	
	• For a given problem, specify the:
	  ○ **Inputs** and the requirements for valid inputs
	  ○ **Outputs** and the requirements for correct outputs
	• Examples:
	  ○ **Business:** Generate an itemized list of purchased items, their costs, and the total payable amount (like a receipt), or calculate mortgage interest and print installment details over time.
	  ○ **Education:** Identify and print the student with the highest score in each subject, or validate user inputs (e.g., test scores); calculate the mean subject grade (MSG) for a class.
	  ○ **Scientific/Mathematics:** Determine if a number is odd or even, or if one number is divisible by another.
	  ○ **Entertainment:** Create a number-guessing game or any game that involves text manipulation.
	• Solve problems by breaking them down into smaller, manageable parts.
	• Identify common patterns across similar problems and make generalizations.
	• Students should be able to identify the individual parts of a problem, create solutions for each part, test the partial solutions, and then combine them to test the overall solution.

7	**Algorithm Design**

	Objectives
	
	This unit focuses on interpreting and understanding algorithms, as well as correcting and writing algorithms for given problems and refining them.
	
	Learning Outcomes
	
	• Perform a dry run of a set of steps to determine its purpose and/or output.
	• Create trace tables to track the values of variables at each stage of a process. This method can be used to identify logic errors or mistakes in faulty algorithms.
	• Identify and locate logic errors in an algorithm, and either correct or modify the algorithm to remove the errors or adapt it to changes in task specifications.
	• Develop an algorithm to solve a problem, presenting it either as a flowchart or in pseudo-code. The following pseudo-code keywords and structures should be used:
	  ○ **INPUT/OUTPUT**
	  ○ **IF… THEN… ELSEIF… ELSE… ENDIF**
	  ○ **WHILE… ENDWHILE**
	  ○ **FOR… NEXT**
	• To facilitate the transition between pseudo-code and Python code. Pseudo-code is not a programming language with a fixed, mandatory syntax. Students' pseudo-code will be evaluated based on the logic of the solution provided. As long as the logic is understood and correctly solves the problem, students will be credited, regardless of whether they adhere to the above style. Pseudo-code with the following conventions:
	  ○ List indices should start from 0
	  ○ Assignment operations should be represented by "="
	  ○ Equality should be represented by "=="
	  ○ Inequality should be represented by "!="

D	Programming
8	**Program Development**

Objectives

This unit focuses on developing programming solutions (coding) for simple problem scenarios. Students will deepen their understanding, with Python being the programming language used in this syllabus.

Learning Outcomes

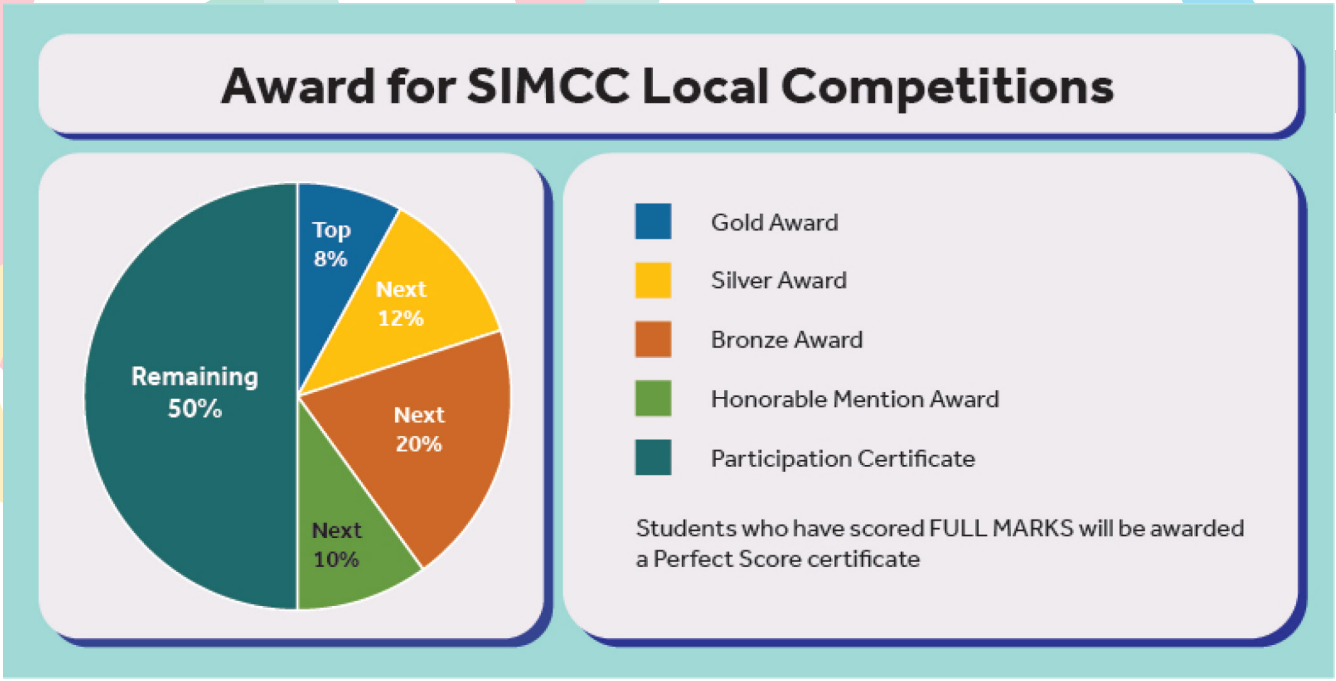
- Understand and describe the stages involved in developing a program: gathering requirements, planning solutions, writing code, testing and refining code, and deploying the code.
- Understand and use sequence, selection, and iteration constructs to create a program.
- Use and justify the use of variables, constants, and simple lists in different problem contexts.
- Python is the programming language for this syllabus. The style and best practices of Python must be followed, such as:
    - Variable names should be in lowercase, with words separated by underscores for better readability.
    - Constants should be written in uppercase, with words separated by underscores.
- Understand and use different data types (integers, floating-point numbers, strings, Booleans, lists) and built-in functions.
- Built-in functions simplify code and can be used in programming solutions. However, their use should not oversimplify the main task of the problem. Problems may specify restrictions on using built-in functions, although this may not always be feasible due to Python's extensive built-in library. For example:
    - To find the smallest number in a list, students should manually write the steps rather than use the built-in min() function.
    - When finding the month with the lowest rainfall from daily rainfall data, students may use min() after calculating and storing monthly totals in a list.
- Develop programming solutions to solve problems such as:
    - Finding the minimum/maximum value in a list
    - Calculating the average of a list of numeric values
    - Searching for an item in a list and reporting the result
    - Finding check digits
    - Determining the length of a string
    - Extracting specific characters from a string

      These skills may include, but are not limited to, comparing items, swapping items, and using mathematical formulas. A string may contain letters, digits, or symbols (e.g., +, :).
- Write and use user-defined functions.

9	Program Testing

	Objectives
	
	This unit focuses on testing and refining programs based on test results. Students will learn how to use test cases effectively and understand which types of testing are necessary and sufficient. Python is the programming language used in this syllabus.
	
	Learning Outcomes
	
	• Identify and justify the use of data validation techniques.
	• Validate input data by performing the following checks:
	  ○ Length check
	  ○ Range check
	  ○ Presence check
	  ○ Format check
	• Design appropriate test cases to cover normal, error, and boundary conditions, specifying what each test case is testing. Boundary conditions refer to the lower and upper limits.
	• Understand and describe different types of program errors: syntax errors, logic errors, and run-time errors, and explain why they occur.
	• Explain how translators are used to detect syntax errors, and describe the difference between an interpreter and a compiler.
	• Understand and apply debugging techniques to isolate, identify, and fix program errors, such as using intermittent print statements or testing the program in small sections or steps.

Awards



The perfect scorer, along with the Gold, Silver, and Bronze medalists of the NJIO national contest, will be invited to join STEAM AHEAD to compete in the STEAM AHEAD 2025 – International Junior Informatics Olympiad (IJIO) for Grades 1 to 10, and the International Junior CyberSecurity Olympiad (IJCO) for Grades 8 and above, organized by the National University of Singapore, School of Computing. Participants will have the opportunity to win the President's Award for Excellence in Steam Star. Students must register with the SIMCC country organizer.

Participation Awards

Schools and participants will receive the following:

1. SGD100 gift voucher each (up to a maximum of $5000) for Perfect Scorers.

2. Medals and Certificates for Perfect Scorers, Gold, Silver and Bronze winners (Except for countries that opt out of medal awards).

3. Personalized medals for Perfect Scorers.

4.E-certificates are also given to participants who qualify for Honorable Mention or attain Certificate of Participation.

5. All participants will receive an Online performance report which analyses their capabilities across different topics and benchmarks their performance with other participants in the same grade and country/territory.

6. For countries with 1,000 or more contestants participating in NJIO 2025, SIMCC will allow our country partners to select the top 6 students to represent their country at the IJIO, free of charge.

Sample Questions

Grades 1 - 2

For more sample questions, visit https://form.simcc.org/lms-home/

Please register an account at our Member Development Portal (https://form.simcc.org/) to access the questions.

Grade 1, BeeBug: A bee and a bug fly together from one flower to the next one from left to right. The flowers are shown in the picture below. If the next flower is higher than the previous flower the bee increases its number by 1. If the next flower is lower than the previous one, then the bug increases its number. If they start with their numbers equal to 0, with what numbers will they end this trip ?

a) Bee – 1, Bug – 1

b) Bee – 2, Bug – 2

c) Bee – 3, Bug – 2

d) Bee – 2, Bug – 3

Answer: c

Grade 2, Swap Sorting: There are seven numbers in a row: 7 6 5 4 3 2 1. Alice wants to have the numbers written in ascending order: 1 2 3 4 5 6 7. She can swap any two numbers having another number between them. For example, she can swap 7 and 5, as there is 6 between them, but she cannot swap 7 and 6 or 7 and 4. What is the minimum number of swaps required to sort the numbers ?

a) 6

b) 7

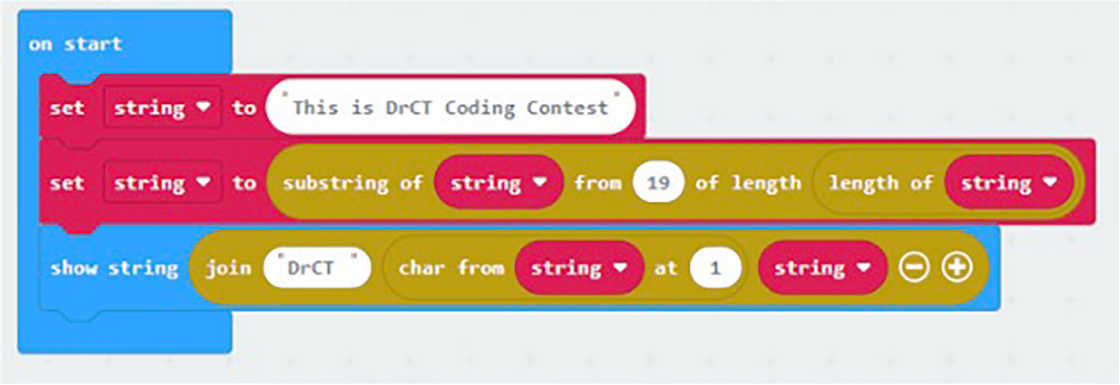
c) 8

d) 9

Answer: d

Sample Questions

Question 3, In NJIO's Junior Coding Club, Alex is learning to code. For his first exercise, he decided to create a manipulate a message using string operations. He used the following code:



Alternative python code:

string = "This is NJIO Coding Contest"

string = string.substr(19, len(string))

basic.show_string("NJIO " + string.char_at(1) + string)

Find out what does the show string block display?

Options

a. NJIO Coding Content

b. NJIO C Contest

c. This is NJIO Coding Contest

d. NJIO Contest

Sample Questions

Grades 3 - 4

For more sample questions, visit https://form.simcc.org/lms-home/
Please register an account at our Member Development Portal (https://form.simcc.org/) to access the questions.

Question 1, Change in the pocket: Liam is planning to go to a store to buy some candies. But he doesn't know how much he will have to pay, and he only likes paying the exact amount. Liam knows that candies can't cost more than $20. There are bills of $1, $5, and $10, as well as coins of 1c, 5c, 10c, 25c, and 50c. So Liam wants to take with him enough bills and coins to pay any price between 1c and $20, but carry as few bills and coins as possible. How many bills and coins in total will Liam have with him?

a) 13

b) 14

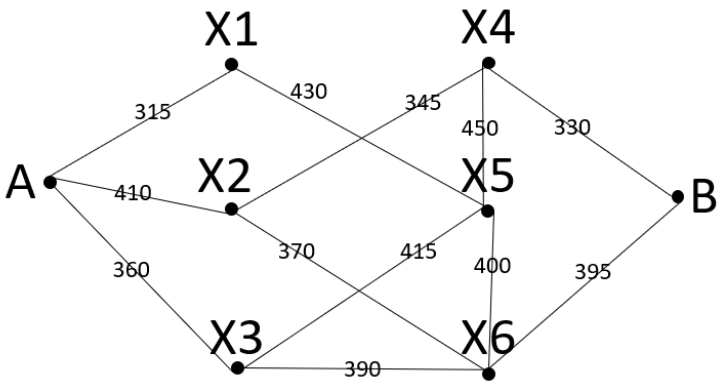
c) 15

d) 16

Answer: c

Question 2, Go through tunnels: You own a delivery company in Switzerland where your trucks deliver goods to customers. But there are a lot of tunnels through the mountains in that country, and every tunnel has a prescribed limit (in cm) on the height of the vehicle that can go through that tunnel. Here is a map of tunnels between two cities A and B.

What is the height of the tallest truck that can go from A to B? (trucks can go using any road available if their height is smaller or equal to a tunnel height limit)



a) 410

b) 370

c) 330

d) 360

Answer: b

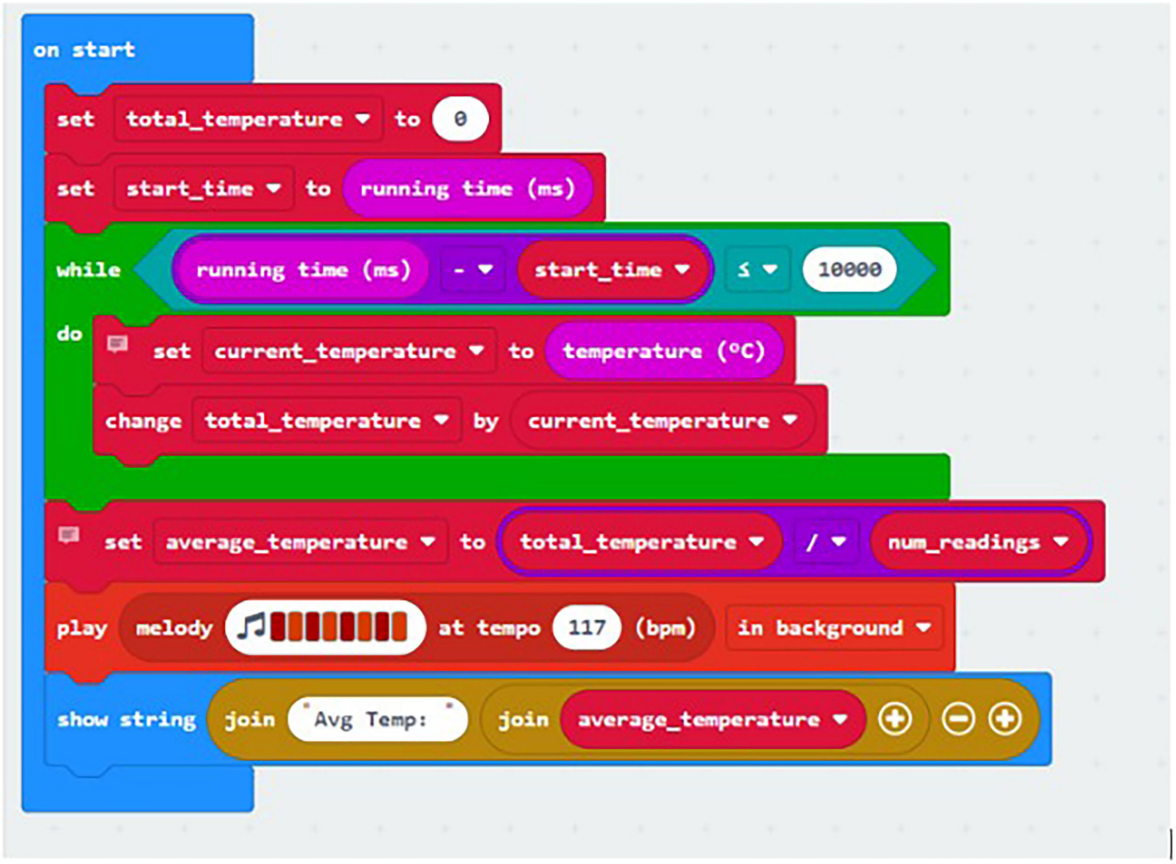
Sample Questions

Question 3

In NJIO's Junior Meteorology Club, students are tasked with creating a program to monitor the average temperature over a certain period using their Micro:bit. The program should measure the temperature every second for 30 seconds and calculate the average temperature. Once the average is calculated, the program should play a melody and display the average temperature on the screen.

Alex wrote the below code, but it is showing the average temperature as infinity. What is wrong in the code. correct it.

Initial Code:



Options:

a. Add basic.pause(1000) inside the while loop.

b. Initialize num_readings to 0 and increment num_readings by 1 inside the while loop with pause of 1 second.

c. Change the division in average_temperature = total_temperature / num_readings to multiplication.

d. Initialize num_readings = 10 before the while loop and pause 1 second inside while loop

Sample Questions

Grades 5 - 6

For more sample questions, visit https://form.simcc.org/lms-home/
Please register an account at our Member Development Portal (https://form.simcc.org/) to access the questions.

Question 1, Ascend: Kyle has the following sequence of numbers and he wants to eliminate as few of them as possible so that all remaining numbers are in increasing order. What is the fewest number of numbers he can eliminate?

5, 6, 10, 7, 19, 25, 3, 44, 24, 72, 17, 31, 5, 42, 28, 56, 69

a) 8

b) 9

c) 10

d) 11

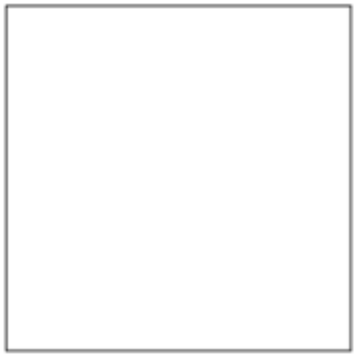
Answer: a

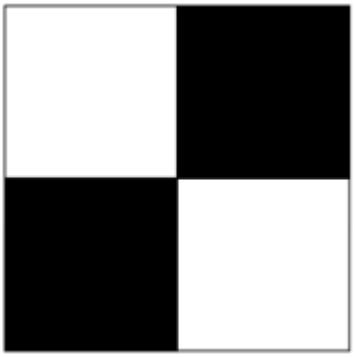
Question 2, Painting squares: Alice has a white square. She divides it into four equal smaller squares and paints lower left and upper right squares black. She repeats this procedure to two smaller white squares, obtaining more smaller white and black squares.

A square screen is initially white. The following procedure is applied to the screen four times:

"Find all white squares on a screen, divide each of them into four smaller squares and paint left lower and right upper small squares black".

How many small white squares will be on the screen in the end? Below is the image of the screen after the first iteration.





a) 14

b) 15

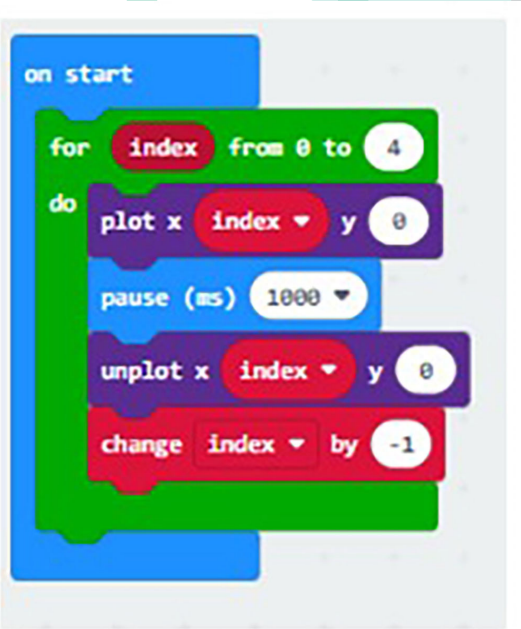
c) 16

d) 17

Answer: c

Sample Questions

Question 3: Azrial is experimenting with for loops on her Microbit LED display to understand how they work. She's trying out this code snippet



What pattern will Azrial see on the Microbit's display when she runs this code?
Options:

a. Display a horizontal line across the top row of the Microbit.

b. Display a vertical line across the left most column of the Microbit.

c. Cause a single LED to blink on and off in the top-left corner five times.

d. Cause a single LED to blink on and off in the top-left corner infinitely.

Sample Questions

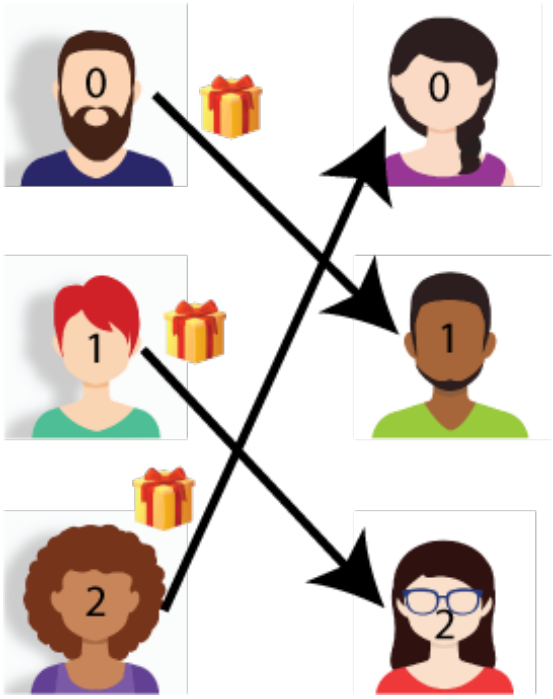
Grades 7 - 8

For more sample questions, visit https://form.simcc.org/lms-home/
Please register an account at our Member Development Portal (https://form.simcc.org/) to access the questions.

Question 1, Gifts: N people, labelled 0, 1, 2, … N - 1 are exchanging gifts. Person i gives a gift to P[i].

Now, each person wants to find out whose gift he received (array Q).

For example, if P = {1, 2, 0}, then Q = {2, 0, 1}.



Explanation: 2 gave his gift to 0, 0 gave his gift to 1, 1 gave his gift to 2.

If P = {4, 1, 5, 2, 8, 9, 7, 3, 6, 0}, find Q.

a) 9, 1, 3, 7, 0, 2, 8, 6, 4, 5

b) 9, 3, 1, 0, 7, 8, 2, 6, 4, 5

c) 5, 1, 3, 7, 0, 2, 8, 6, 4, 9

d) 5, 3, 1, 0, 7, 2, 8, 6, 4, 9

Answer: a

Question 2, log2: For example, log2(2) = 1. Since 2/2 = 1. log2(7) = 2 since 7/2 = 3.5, 3/2 = 1.5.

What is the value of log2(2020)?

a) 9

b) 10

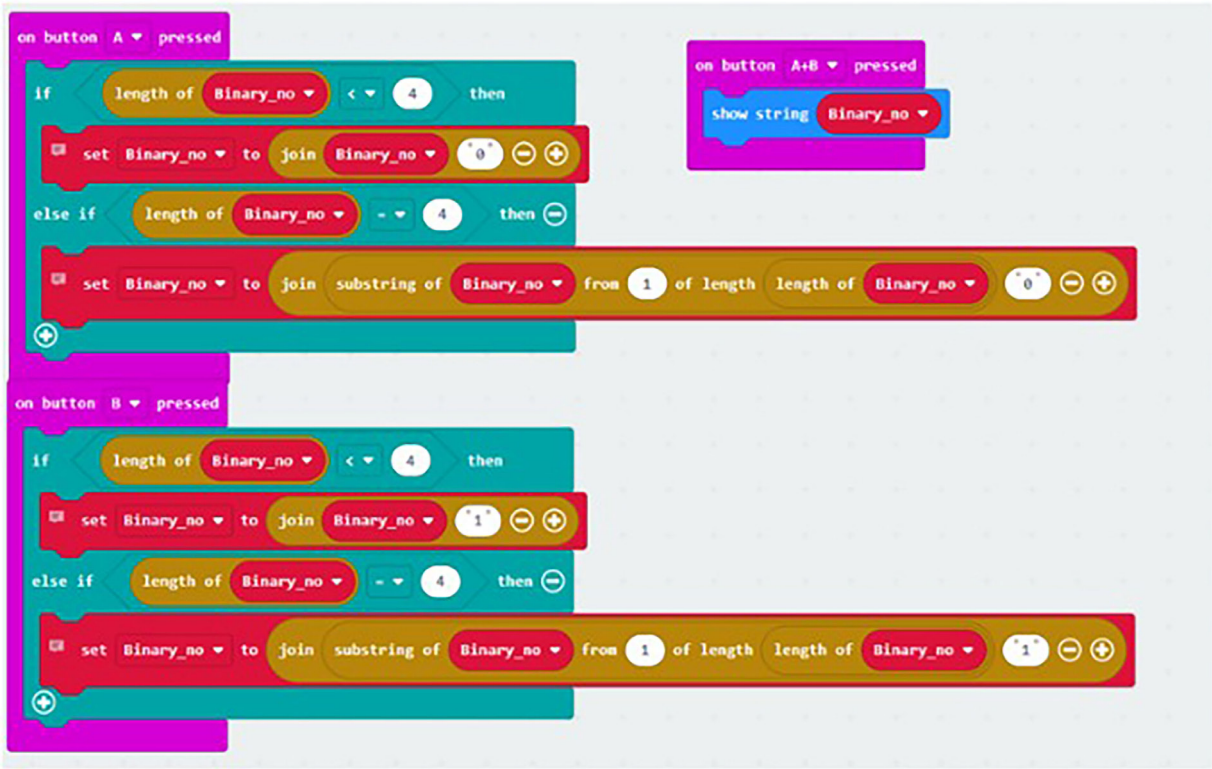
c) 11

d) 20

Answer: b

Sample Questions

Question: 3: Sophie coded the following program in Microsoft MakeCode blocks to experiment with binary numbers. She initializes an empty binary number Binary_no and adds digits to it using buttons A and B. Pressing button A adds "0" to the binary number, while pressing button B adds "1". When both buttons A and B are pressed together (Button A+B), the micro:bit displays the current binary number on its LED display.



What would be displayed on the micro:bit's LED display after Sophie performs the following sequence of button presses:

1. Presses button A three times.
2. Presses button B four times.
3. Presses buttons A and B together (Button A+B).

Options:

a. "0001111"

b. "0001"

c. "0011"

d. "1111"

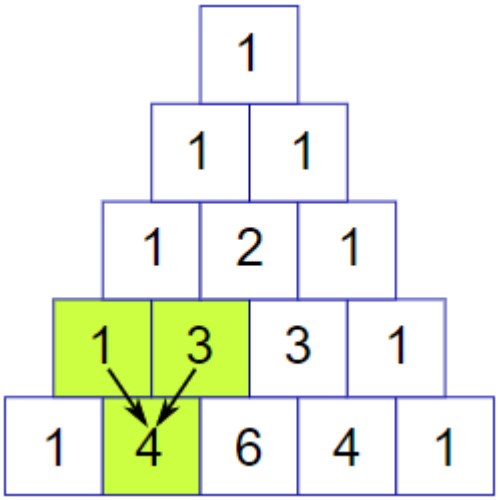
Sample Questions

Grade 9 and above

For more sample questions, visit https://form.simcc.org/lms-home/
Please register an account at our Member Development Portal (https://form.simcc.org/) to access the questions.

Question 1, Pascal: The rows of Pascal's triangle start with row n = 0 at the top (the 0th row). The entries in each row are numbered from the left beginning with k = 0 and are usually staggered relative to the numbers in the adjacent rows. For example, the number at n=4, k=1 is 4.

The triangle may be constructed in the following manner: In row 0, there is an entry of 1. Each entry of each subsequent row is constructed by adding the number above and to the left with the number above and to the right, treating blank entries as 0.



What is the number at n=12, k=5?

a) 970

b) 792

c) 972

d) 729

Answer: b

Question 2, findnumber: I am thinking of an integer x from 1 to 100. You want to find out what x is by asking questions that go like: "is x greater than y?", where y is an integer of your choice.

What is the minimum number of questions you must ask to guarantee you can find the correct value of x?

a) 5

b) 6

c) 7

d) 8

Answer: c

Sample Questions

Question 3: Jason is enhancing his website's user experience by integrating a sophisticated field validator using the Google Blockly API. He aims to create a validator that not only filters out specific characters from the text input but also implements a complex validation rule: it should reject any input where the number of digits exceeds the number of alphabetical characters. To achieve this, Jason decides to write a new Blockly block that includes a custom validator.

Examine the code snippet Jason is considering for this purpose:

```
Blockly.Blocks['advanced_validator'] = {
  init: function() {
    var advancedValidator = function(newValue) {
      // Custom validation logic will be inserted here
    };

    this.appendDummyInput()
        .appendField(new Blockly.FieldTextInput('Enter text', advancedValidator));
  }
};
```

Which among the following options should Jason insert as the custom validation logic inside the advancedValidator function to meet his objective?

Options:

A.

```
var digitCount = (newValue.match(/\d/g) || []).length;
var charCount = (newValue.match(/[a-zA-Z]/g) || []).length;
return digitCount > charCount ? '' : newValue;
```

B.

```
var digitCount = (newValue.match(/\d/g) || []).length;
return digitCount === 0 ? newValue : '';
```

C.

```
var charCount = (newValue.match(/[a-zA-Z]/g) || []).length;
return charCount > 5 ? '' : newValue;
```

D.

```
var specialChars = (newValue.match(/[^a-zA-Z0-9]/g) || []).length;
var totalCount = newValue.length;
return specialChars > totalCount / 2 ? '' : newValue;
```

Registration Information

Competition Dates

Registration Fees

Platform

Online and
Paper base

HOW TO REGISTER

Kindly check with your country partner for registration and competition details. For more information about your country partner, please refer to our website below

https://simcc.org/country-partners/

Refund Policy

The contest fees paid by students to the competition are non-refundable. To host the competition, our organization invests a significant amount of time and resources, not to mention the various charges incurred to process the payments and registration.

As a social enterprise, SIMCC operates with a very lean team and limited resources to keep our operating costs low in order to make our competition affordable to all students. Hence, we will not be able to offer any refunds for competition fees to students who withdraw or cancel beyond our control.

If any student has been wrongly charged by SIMCC, or we cancel an event due to reasons beyond our control, we will happily refund the fees paid by the students.

STEAM AHEAD

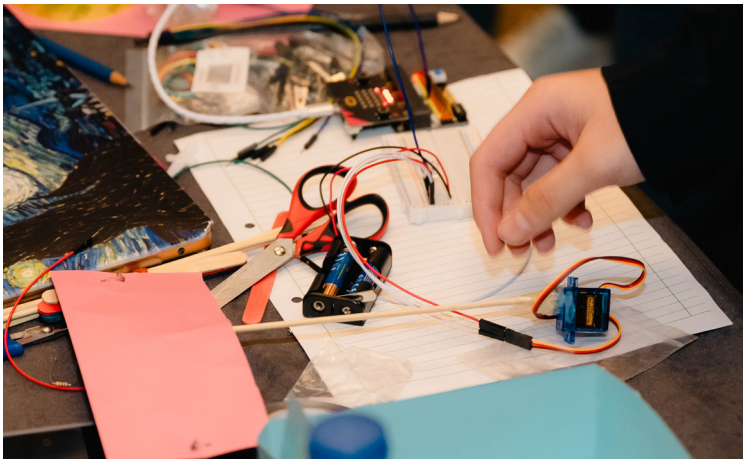
A COMBINATION OF 5 COMPETITIONS: International Junior Math Olympiad (IJMO), VANDA International Science Olympiad(VIJSO), International Junior Informatics (IJIO) and International Junior Arts Olympiad(IJAO)), International Junior CyberSecurity Olympiad. (IJCO) The National University of Singapore Centre for Nurturing Computing Talent (NUS CeNCE) and Singapore's National Cybersecurity R&D Lab (NCL) will be offering Capture the Flag (CTF) as part of the International Junior CyberSecurity Olympiad (IJCO). Online Video training recorded will be provided by NUS CeNCE and NCL professors and the contest certificates and medals will be provided by NUS and NCL.

STEAM stands for Science, Technology, Engineering/Entrepreneurship, Arts and Mathematics. STEAM AHEAD is our initiative to combine our international academic competitions to educate students and bring them international exposure about possible career choices in these fields.

So, STEAM AHEAD offers multiple opportunities for students to win awards in IJMO Individual, VIJSO Individual & Team, and IJIO & IJCO Individual & Team, together with IJAO and IJIO Individual, plus Overall Champion in each grade level for IJMO, VIJSO, IJIO and IJCO, and 4 individual Gold Medal winners in IJMO, IJCO, VIJSO, IJIO and IJAO will be awarded the President's Award for Excellence in STEAM STAR (PAExS STAR) Students winning 3 individual gold medals in any of the STEAM AHEAD Competitions will be awarded President's Award for Excellence in STEAM (PAExS)

Please refer to the STEAM AHEAD info pack for more details or visit our website.

WEBSITE: https://ijmo.asia/



International Junior Honor Society

"Leaders Give, Givers Get!"

IJHS is an honor society that focuses on developing leaders and enabling its members to achieve success in academic and life pursuits. IJHS provides its members with a variety of platforms and opportunities to unleash their full potential, as well as connect to their community.

IJHS is fully funded by Scholastic Trust Singapore (STS), a non-profit foundation, and supported by a prominent volunteer board of advisors to help guide bright young leaders.

About IJHS

- **Vision**: a strong community of compassionate leaders.
- **Values**: humility, empathy, adaptability, resolute, truthfulness.

Goals

- Unlock the potential, talents, passions, and interests of aspiring Fellows
- Empower young leaders to give back to their local and global communities
- Pave the way for members to gain entry into top schools and qualify for scholarships
- Provide top institutions' admission and scholarship coaching to high achievers
- Make the world a better place.

Why should you join IJHS?

01 Expert guidance on how to gain entrance into renowned educational institutions/universities.

02 Important internship opportunities where members can explore a variety of career options while enhancing their leadership skills and admission/scholarships at top universities.

03 Myriad resources and support to enable members to contribute back to their community.

04 Mentorship to assist members in qualifying for prestigious scholarships.

05 Access to a global network of like-minded Leaders and Givers.

How to earn IJHS SPs?

Students earn IJHS Scholarship points (SPs) by winning awards in SIMCC competitions. There are 3 levels of SIMCC competitions with different SPs.

Individual contests	National Contest	Global Finals Online	Global Finals
Perfect Score	3	3.5	4
Gold Award	2	2.5	3
Silver Award	1	1.5	2
Bronze Award	0.5	0.75	1

Team contests	MMT	VIJSO	IJCO	IJIO	SIMOC
Gold Award		2			1
Silver Award		1			0.5
Bronze Award		0.5			0.25

IJHS Membership Level

04 **Honorary Lifetime**
Special membership award for those who contribute to the society

03 **Lifetime**
3rd Entry Lifetime membership, tracked by LT Year # with LT Year 10 being the highest honor.

02 **Senior**
2nd Entry
2 years membership

01 **Junior**
1st year Entry
1 year membership

Other Awards counted for IJHS Individual SPs:

STEAM International Tournament (SIT), PAExS, Gold, Silver, Bronze awards in SIMCC contests plus Overall Championships, and YALA.

YALA Scholarships

YALA
YOUNG ACHIEVERS LEADERSHIP ACADEMY

Every member of IJHS (grades 4 and above) will automatically receive a scholarship to join the YALA Camp every year.

There are 4 levels of scholarships based on IJHS Scholarship Points (SPs).

1. **Platinum Scholarship (100%)**
2. **Palladium Scholarship (75%)**
   New class of scholarships for IJHS members to apply for YALA scholarships
3. **Gold Scholarship (50%)**
4. **Silver Scholarship (30%)**
   Open to all IJHS members to apply

IJHS
INTERNATIONAL JUNIOR HONOR SOCIETY

Gained >= 4 IJHS SPs → **Member of IJHS**

Gained more IJHS SPs → **Better and more scholarships**

Young Achievers Leadership Academy (YALA) Camp

Watch a video of YALA 2023 hosted in Bali, Indonesia and see the exciting skills and recognition IJHS members have acquired through the YALA Scholarship and how they have contributed back to their local and global communities through the Community Support Projects (CSPs).
https://simccorg.sharepoint.com/:v:/s/CompetitionContestPapersCountrypartnersandPrintedversion/EWTGH59emOhDkW_Xfs3iUgIBPxt_615k1oRCQihWkFFyrA?e=FcX6ZK

Singapore International
Mastery Contests Center