

Computational Thinking & Data Representation Cheat Sheet (Theory Only)

Computational Thinking

- A **problem-solving approach** involving:
 - A **Decomposition** → Breaking problems into smaller parts.
 - B **Pattern Recognition** → Identifying similarities in problems.
 - C **Abstraction** → Focusing on relevant details.
 - D **Algorithm Design** → Creating step-by-step solutions.
-

Information & Data

- **Information** → Processed and meaningful data.
 - **Data** → Raw facts and figures without meaning.
 - **Example:**
 - **Data:** 23, 45, 67
 - **Information:** "The average test score of students is 45."
-

Converting Information into Data

- **Qualitative Data** → Text, images, sounds (converted into numeric form).
 - **Quantitative Data** → Numerical values, measurable data.
 - **Example:**
 - **Information:** "The temperature is cold."
 - **Data:** Temperature = 10°C.
-

Data Capacity

- **Data is stored in binary (0s & 1s).**
 - **Memory Size Units:**
 - 1 Byte = 8 bits
 - 1 Kilobyte (KB) = 1024 Bytes
 - 1 Megabyte (MB) = 1024 KB
 - 1 Gigabyte (GB) = 1024 MB
 - 1 Terabyte (TB) = 1024 GB
-

Data Types & Encoding

1st Data Types in Programming:

- **Integer (int)** → Whole numbers.
- **Float (float)** → Decimal numbers.
- **Character (char)** → Single character ('A').
- **String (char[])** → Sequence of characters ("Hello").
- **Boolean (bool)** → True or False.

1st Encoding Methods:

- **ASCII** → 7-bit encoding for characters.
- **Unicode** → Universal character encoding (UTF-8, UTF-16).
- **Binary Encoding** → Stores numbers as 0s and 1s.

Logic & Problem Solving

- **Boolean Logic** → Uses **AND**, **OR**, **NOT** operators.
- **Truth Table Example:**

A	B	A AND B	A OR B	NOT A
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Limits of Computation

- **Decidability** → Some problems **cannot** be solved by algorithms.
- **Complexity** → Measures the efficiency of an algorithm.
- **Examples:**
 - **P vs NP Problem** → Determines if complex problems can be solved efficiently.
 - **Halting Problem** → Some programs **cannot** determine if another program will stop.

Pseudocode & Flow Chart

Pseudocode

- A **high-level description** of an algorithm using simple statements.

Example (**Find the largest number**):

Start

Input A, B

If A > B then

 Print "A is larger"

Else

 Print "B is larger"

End

Flow Chart Symbols

Symbol	Description
Oval	Start/End
Rectangle	Process (Calculation)
Diamond	Decision (If/Else)
Parallelogram	Input/Output

Example Flowchart for Finding Largest Number:

Start

↓
Input A, B
↓
A > B? → Yes → Print "A is larger"
↓ No
Print "B is larger"
↓
End

This **Computational Thinking Cheat Sheet** covers **data representation, problem-solving, logic, computation limits, pseudocode, and flowcharts**. Let me know if you need more details!