
Computational thinking
Problem solving and programming

# Lecture 1

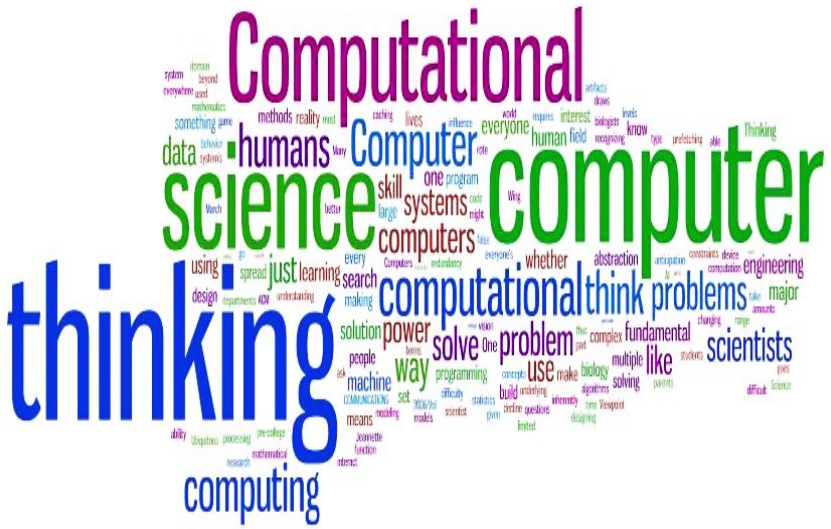
# Introduction to computational thinking

# LECTURE OBJECTIVES

- Define computational thinking
- Computational thinking skills:
  - Decomposition
  - Abstraction
  - Pattern recognition
  - Algorithmic thinking
- Writing an algorithm for som practical problems

# Computational Thinking (CT)

- CT Computational thinking (CT) is a problem-solving approach using computer, that formulates a problem and its solution to be effectively executed by a computer. (Wing 2014).
- CT is a problem-solving approach where the ultimate aim is to provide a solution whose form is ready to be programmed into a computer.



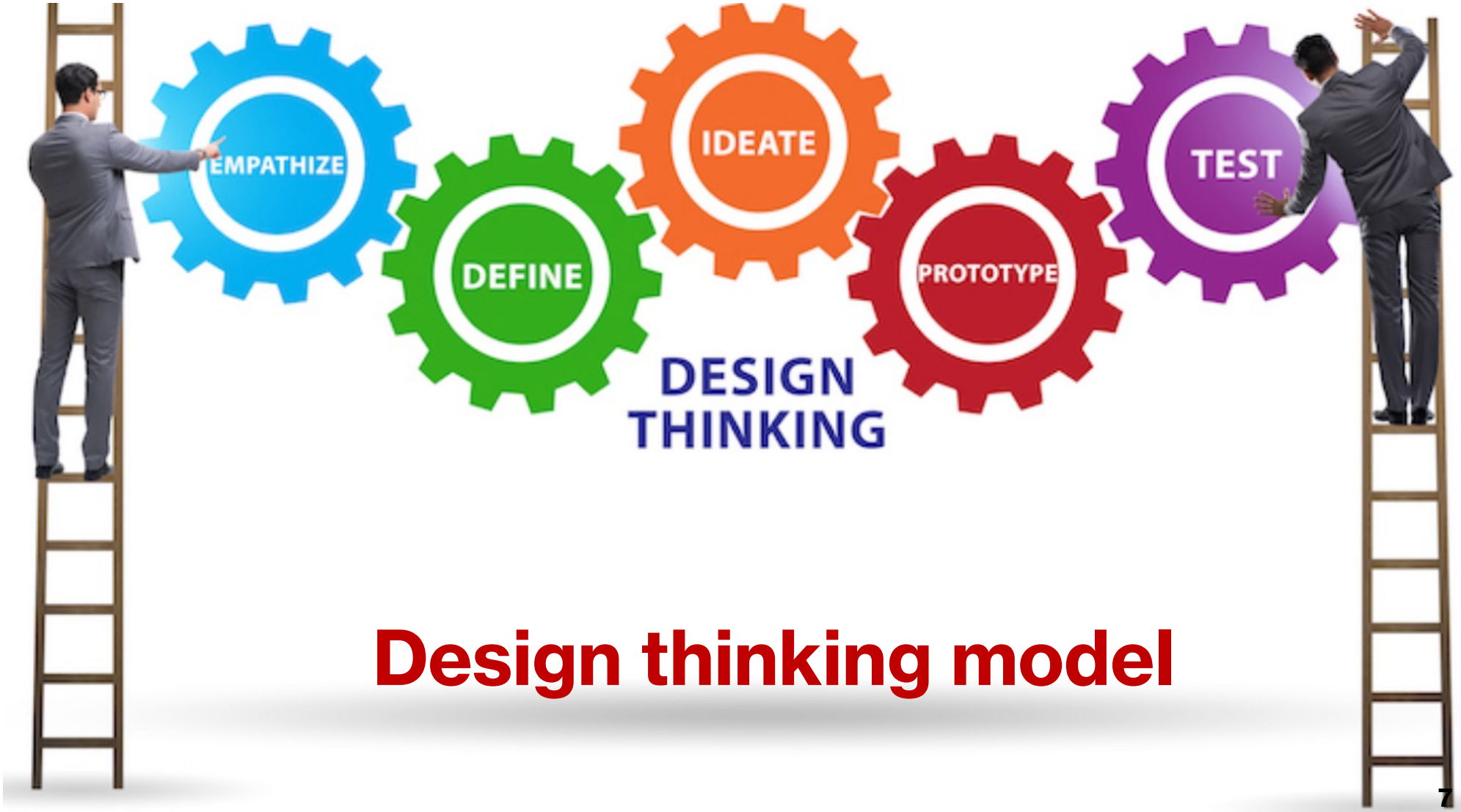
# How is computational thinking used?

- Anyone can apply CT when solving a problem and have a computer play a role in the solution:
  - Mathematician: carry out long division factoring or doing carries in addition or subtraction.
  - Scientist: do an experimental procedure.

# Design thinking model

It was designed by the Hasso-Plattner Institute of Design at Stanford University. The main steps of the model is:

1. **Empathize**: Understand the problem from the audience.
2. **Define**: Identify the objectives and the decisions that need to be made.
3. **Ideate**: Brainstorm ideas.
4. **Prototype**: Design the algorithm solution and check it often.
5. **Test**: Check your algorithm often throughout the process and go back to previous steps as needed.




# Design thinking model

## Some key advantages of developing computational thinking skills:

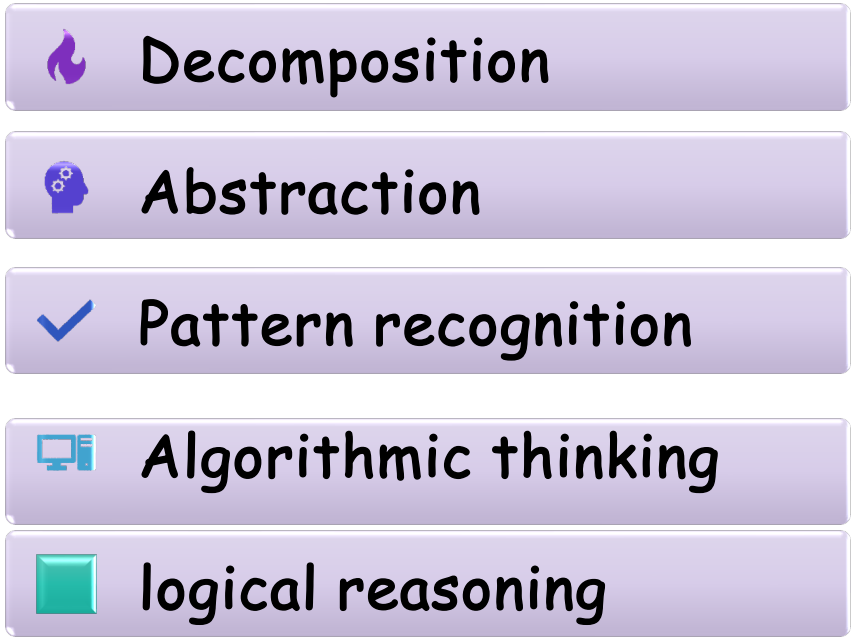
1. **Problem-solving skills**: CT equips individuals with a structured approach to problem-solving. It emphasizes breaking down complex problems into smaller, more manageable parts, identifying patterns, and designing algorithms to solve them.
2. **Logical reasoning**: CT promotes logical reasoning by encouraging individuals to analyze problems, identify cause-and-effect relationships.
3. **Creativity and innovation**: CT encourages individuals to think creatively and come up with innovative solutions.

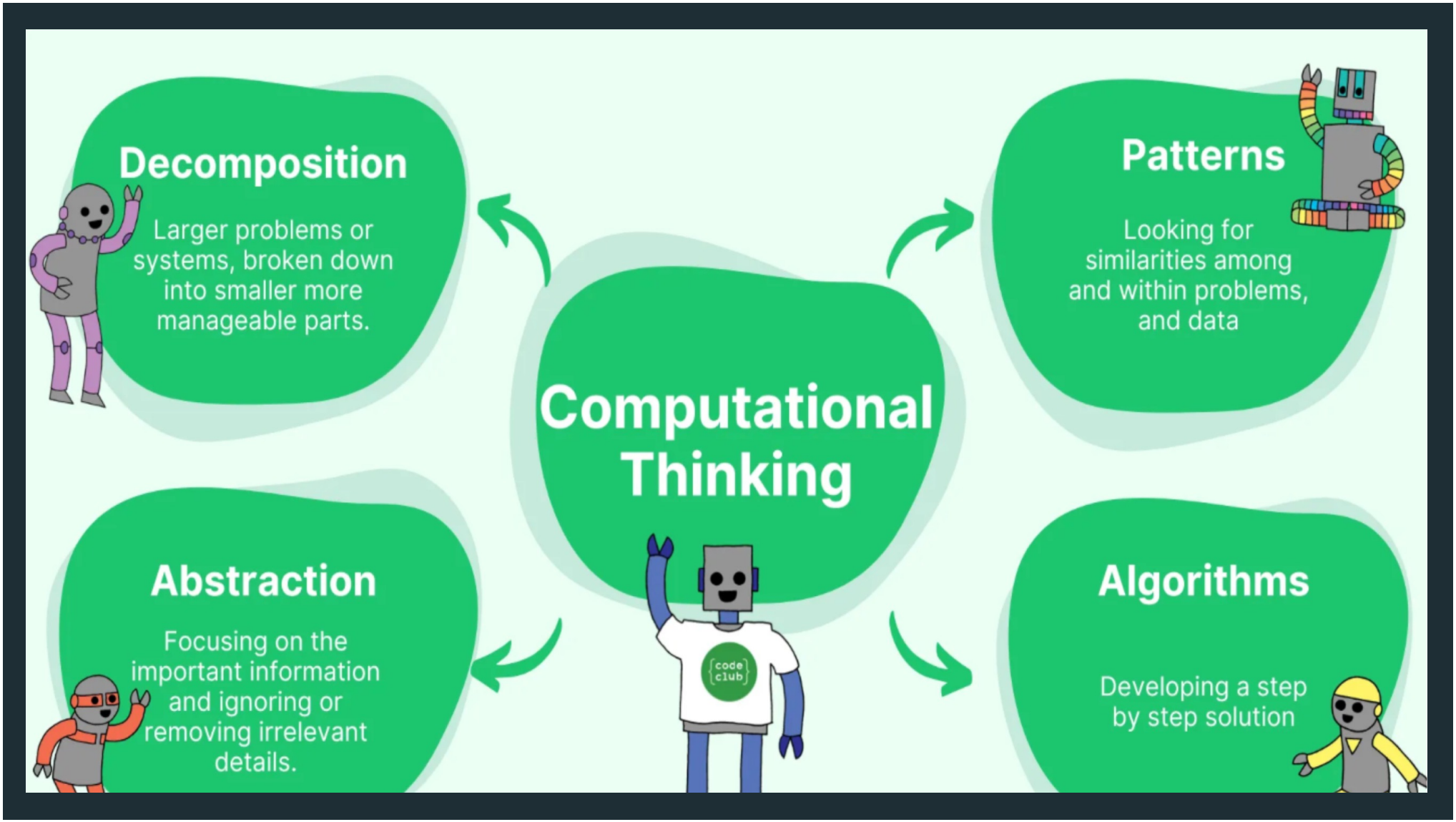
## Some key advantages of developing computational thinking skills:

4. **Data analysis and interpretation**: CT helps individuals understand how to collect, organize, and analyze data.
5. **Algorithmic thinking**: CT fosters the ability to design and implement algorithms. Algorithms are step-by-step instructions for solving problems.
6. **Collaboration and teamwork**: CT often involves working collaboratively on problem-solving tasks.
7. **Transferable skills**: CT can enhance critical thinking, problem-solving and analytical skills in various fields, including business, healthcare, finance.

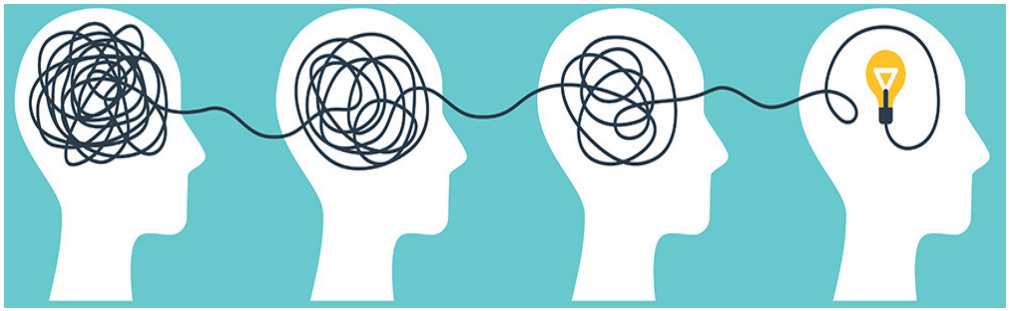
# Computational thinking pillars (skillls)

- These characteristics will help you to think computationally through a complex problem:




Decomposition
Larger problems or systems, broken down into smaller more manageable parts.
Patterns
Looking for similarities among and within problems, and data
Computational Thinking
Abstraction
Focusing on the important information and ignoring or removing irrelevant details.
Algorithms
Developing a step by step solution
code club

# 1. Decomposition



Decomposition is a process of breaking down a big problem into small and manageable parts.

The smaller parts can then be solved by the computer, as they are simpler to work with.

"If you can't solve a problem, then there is an easier problem you can solve: find it" George Polya.

# Decomposition (Example)

Look at an example, the equation to find the roots of a quadratic equation:

$$ax^2 + bx + c = 0$$

$$\Rightarrow \quad x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

# Decomposition (Example)

- On first look, it might appear a little scary, but if we decompose it, we should stand a better chance of solving it:

1) $b^2$
2) $4ac$
3) $b^2 - 4ac$
4) $\sqrt{b^2 - 4ac}$
5) $-b \mp \sqrt{b^2 - 4ac}$
6) $2a$
7) $x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$
8) $x = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$

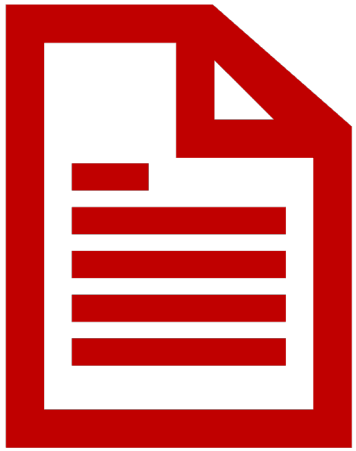
# Decomposition (Example)

Identify all of the tasks you will need to do in order to bake a cake.

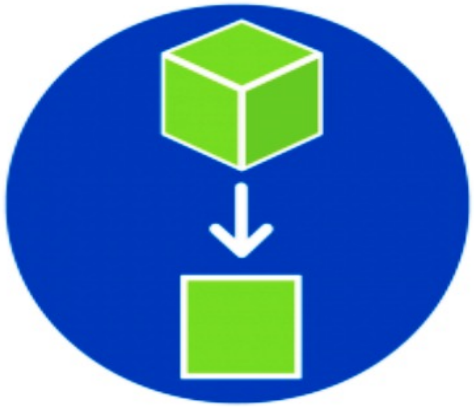
Create a video game with your friends. How could you divide up the tasks?

Create a code that will convert a value in Fahrenheit to one in Celsius. What are the major parts and processes?

# 2. Abstraction



- **Abstraction** is about reducing the complexity of a problem or task by focusing on what is important and removing unnecessary details.
- It can also be used to have one object stand for many, or to have a word stand for an action. Models and simulations can also be considered abstractions.

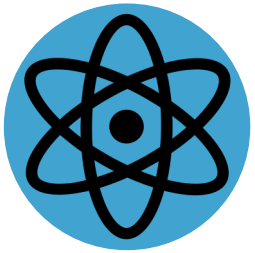


# Abstraction (Example)

Find the important skills for a baseball pitcher.

Represent the actions run, stop and hop using images.

Code a simulation of a volcano erupting.



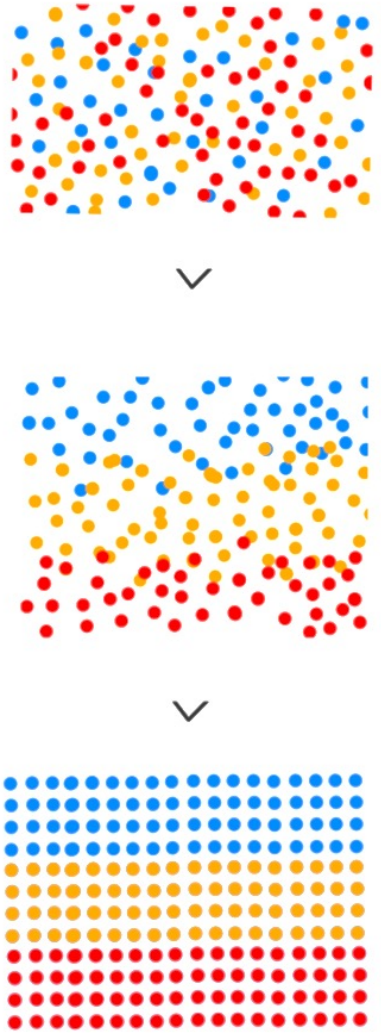
Find the appropriate formula to solve a physics problem.

# 3. Pattern recognition

- It helps to recognize patterns to describe and represent sequences in data or processes.
- Find the pattern behind this data:

  1, 4, 7, 10, 13, 16, 19, 22, 25, ...

- Pattern recognition might also involve recognizing shapes, sounds or images.

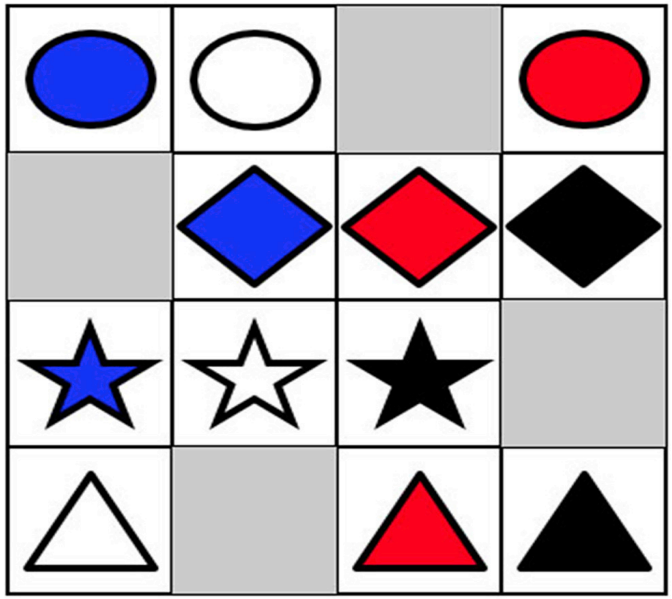


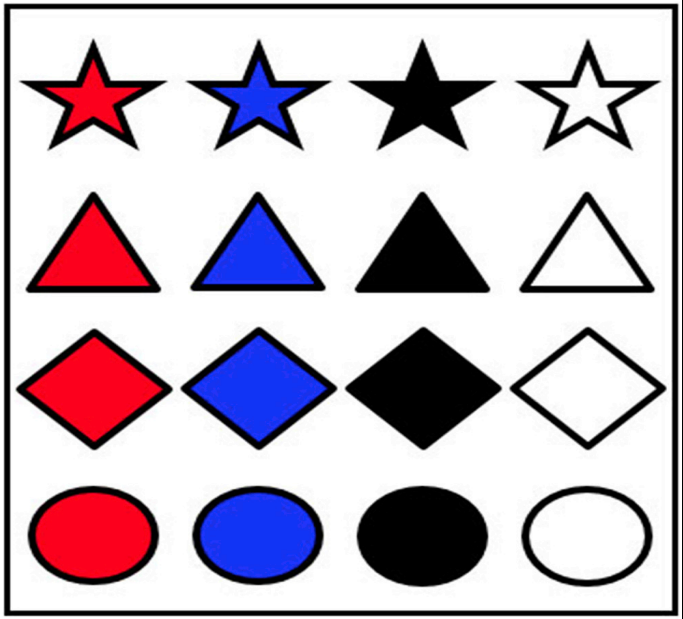
# Pattern recognition (Example)

Drag objects from the box on the right into the gray cells to complete the pattern.

Each colored shape can only appear once in your solution.

Click “Submit” when you think you have it right.





# 4. Algorithmic thinking

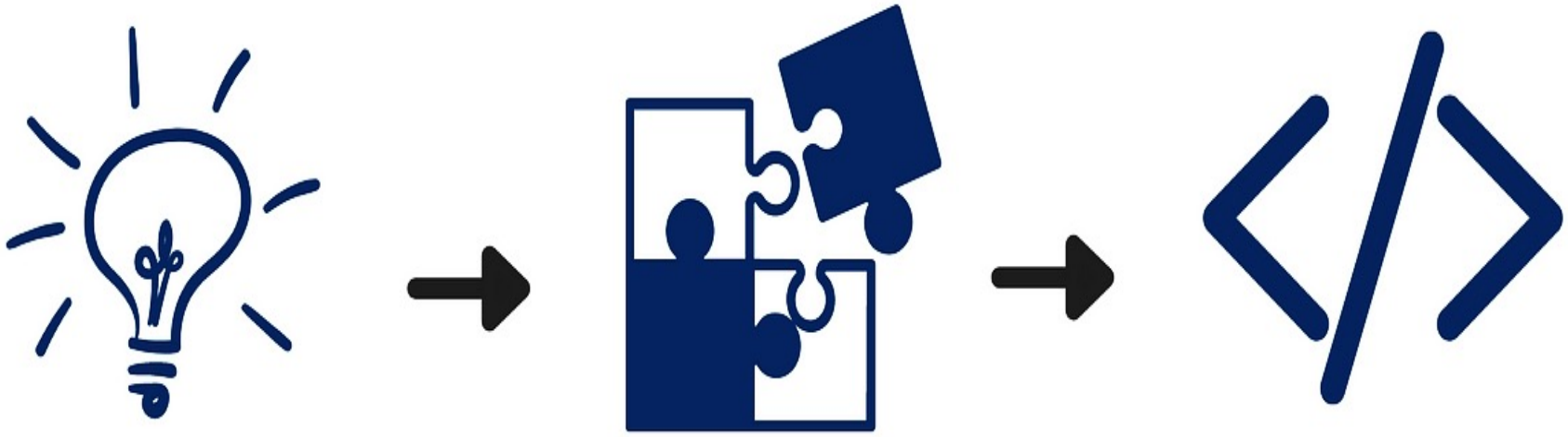
- **Algorithm** is a series of ordered and unambiguous instructions to solve a problem.
- Computer scientists seeks to find an effective and efficient algorithm to solve a problem with minimum computing resources (memory or time) while getting the correct output.

# Algorithm representation

- The algorithm can be represented using:
  - **Pseudocode**
  - **flowchart**

# How to develop Algorithmic Thinking?

enjoy algorithms



Solution Idea

Efficient Algorithm

Correct Code

# Algorithmic thinking (Example)

- What steps are needed to make a sandwich?
- Does it matter if the steps are done in a different order? Why or why not?
- Create an algorithm to teach a young child how to tie shoelaces in a bow.

# Characteristics of an algorithm

**An algorithm must possess following characteristics :**

1. **Finiteness**: An algorithm should have finite number of steps and it should end after a finite time.
2. **Input**: An algorithm may have many inputs or no inputs at all.
3. **Output**: It should result at least one output and correct.
4. **Definiteness**: Each step must be clear, well-defined and precise. There should be no any ambiguity.
5. **Effectiveness**: Each step must be simple and should take a finite amount of time.

# Guidelines for developing an algorithm

1. An algorithm will be enclosed by START (BEGIN) and STOP (END).

2. To accept data from user, generally used statements are INPUT, READ, GET or OBTAIN.

3. To display result or any message, generally used statements are PRINT, DISPLAY, or WRITE.

4. Generally, COMPUTE or CALCULATE is used while describing mathematical expressions and based on situation relevant operators can be use.

# Advantages of an algorithm

1. **Effective Communication**: Since algorithm is written in English like language, it is simple to understand step-by-step solution of the problems.
2. **Easy Debugging**: Well-designed algorithm makes debugging easy so that we can identify logical error in the program.
3. **Easy an Efficient Coding**: An algorithm acts as a blueprint of a program and helps during program development.
4. **Independent of Programming Language**: An algorithm is independent of programming languages and can be easily coded using any high-level language.

# Office lunch Python algorithm

Problem 1. Determine the final cost for an office lunch for employees given two possible options:

- $8.50 for a sandwich meal
- $7.95 for a salad meal

The mathematical equation for the total cost is:

$$Cost = 8.5No\_sandwiches + 7.95No\_salad$$

```
How many sandwich lunches were ordered? 12
How many salad lunches were ordered? 23
The total cost for the employee lunch is $284.85.
```

# Office lunch Python algorithm (Pseudocode)

**Step 1**: Start
**Step 2**: Read number of sandwiches No_sandwiches
**Step 3**: Read number of salad lunches No_salad
**Step 3**: Calculate Cost = 8.5No_sandwiches + 7.95No_salad
**Step 4**: Print cost
**Step 5**: stop

# Algorithm

We want to ask ourselves some questions:

- Does the algorithm do what we want?
- Does the output make sense?
- Is there another way to get the same information in a clearer way?

# Algorithm for adding two numbers

**Step 1**: Start

**Step 2**: Read two numbers a and b

**Step 3**: Calculate Sum = a + b

**Step 4**: Print sum

**Step 5**: stop

# Example of an algorithm

**Algorithm** : Calculation of Simple Interest

**Step 1**: Start

**Step 2**: Read principle (P), time (T) and rate (R)

**Step 3**: Calculate I = P*T*R/100

**Step 4**: Print I as Interest

**Step 5**: Stop

Create an algorithm to compute the volume of a sphere. use the formula: $v = (4/3) * pi * r^3$ where pi is equal to 3.1416 approximately. the r is the radius of sphere. display the result. (Design the flowchart)

**Step 1**: Start

**Step 2**: Read r

**Step 3**: Calculate $v = (4/3) * pi * r^3$ .

**Step 4**: Print v

**Step 5**: Stop

Write an algorithm that converts the input Celsius degree into its equivalent Fahrenheit degree. Use the formula: F = (9/5) *C+32.

**Step 1**: Start

**Step 2**: Read C

**Step 3**: Calculate F = (9/5) *C+32.

**Step 4**: Print F

**Step 5**: Stop

## Algorithm for find the greater number between two numbers.

**Step 1**: Start

**Step 2**: Read A and B

**Step 3**: if A>B.

**Step 4**: Print A

Step 5: Otherwise, print B

**Step 5**: Stop

**Problem 3**: Write an algorithm to find the average of temperature T1, T2, and T3.

Inputs: The temperature T1, T2, And T3.

Expected output: The average of three temperature.

Step1: Start

Step2: Read the values f temperature T1, T2, and T3..

Step3: Calculate the average=(T1+T2+T3)/3

Step4: Print the average

Step5: End

**Problem 3**: Write an algorithm to find the average of temperature T1, T2, and T3.

Inputs: The temperature T1, T2, And T3.

Expected output: The average of three temperature.

Step1: Start

Step2: Read the values f temperature T1, T2, and T3..

Step3: Calculate the average=(T1+T2+T3)/3

Step4: Print the average

Step5: End

**Problem 4:** write algorithm to find the greater number between two numbers

Step1: Start

Step2: Read/input A and B

Step3: If A greater than B then C=A

Step4: if B greater than A then C=B

Step5: Print C

Step6: End

**Problem 5:** Write An algorithm to calculate even numbers between 0 and 99

1. Start
2. I = 0
3. Print I in standard output
4. I = I+2
5. If (I <=98) then go to line 3
6. End

# Questions

1. Define Computational thinking.
2. State the computational thinking skills.
3. List the core concepts of CT.

## Write an algorithm for

$$A = P\left(1 + \frac{r}{n}\right)^{nt}$$

- $A$ is the total amount.
- $P$ is the principal amount, that is, the initial deposit.
- $r$ is the interest rate (keep in mind that for 3%, the interest is written as 0.03, for example).
- $n$ is the number of times interest is compounded per year.
- $t$ is the number of years the deposit goes untouched.

## Questions

- Write an algorithm to compute the radius of a circle. Derive your formula from the given equation: $A=\pi r^2$, then display the output
- Determine the most economical quantity to be stocked for each product that a manufacturing company has in its inventory: This quantity, called economic order quantity (EOQ) is calculated as follows: EOQ=2rs/1 where: R= total yearly production requirement S=set up cost per order I=inventory carrying cost per unit.
- Write an algorithm that takes as input the purchase price of an item (P), its expected number of years of service (Y) and its expected salvage value (S). Then outputs the yearly depreciation for the item (D). Use the formula: D = (P – S) Y.

# Questions

- Design an algorithm to find the circumference of a circle. Use the formula: $C=2\pi r$, where $\pi$ is approximately equivalent 3.1416.
- Write an algorithm that converts an input inch(es) into its equivalent centimeters. Take note that one inch is equivalent to 2.54cms.
- Write an algorithm that converts the input dollar to its peso exchange rate equivalent. Assume that the present exchange rate is 51.50 pesos against the dollar. Then display the peso equivalent exchange rate.