

Introduction to programming

Musaeus College
Colombo



Programming

Unit 3

Grade 9 - ICT

20/07/2020



An Algorithm

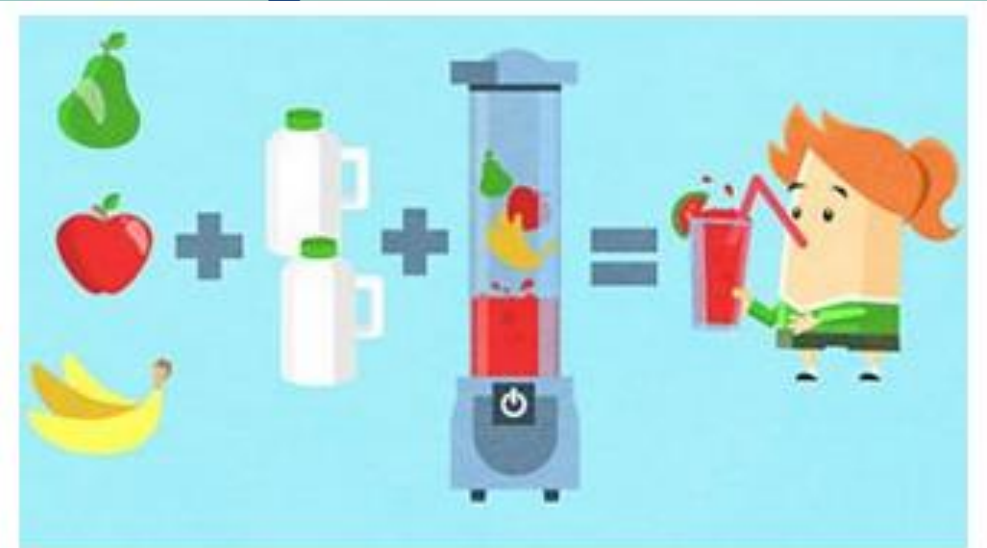


- Simply a sequence of steps for completing a task.
- An algorithm is a sequence of instructions or a set of rules that are followed to complete a task.
- This task can be anything, so long as you can give clear instructions for it.
- For example, to make yourself a drink you have to follow a sequence of steps in the right order. If you do something in the wrong order you might end up making a mess.

An Algorithm ...



- When you write an algorithm you need to include precise, step-by-step instructions.
- Here are the instructions for making a smoothie:
 1. Add fruit to the blender.
 2. Add milk to the blender.
 3. Put the lid on the blender.
 4. Switch the blender on.



An algorithm is a clear set of instructions. In this case the instructions are how to make a smoothie.

Examples



- Write an algorithm to make a fruit salad.
- Write an algorithm for getting ready to go to school.

Making a fruit salad



1. Wash fruits
2. Peel off fruits
3. Cut into pieces
4. Add some sugar
5. Add pinch of salt
6. Mix together



Getting ready to go to school



1. Wake up in the morning
2. Brush the teeth
3. Take a wash
4. Take the breakfast
5. Get dressed with the uniform
6. Wear shoes
7. Comb hair

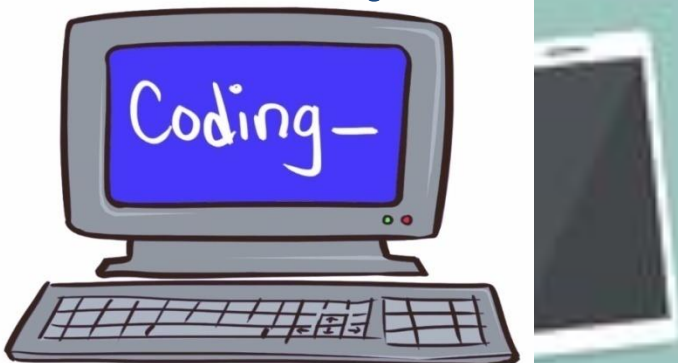


You know that you put on your underwear first and your clothes afterwards. If you do those tasks in the wrong order you might end up looking like a superhero!

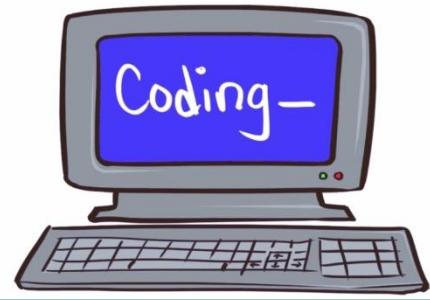
Speaking computer's language



- Computers won't understand algorithms as they use a different language.
- It will need to be translated into a code which the computer will then follow to complete a task.
- A computer program(code) is a series of algorithms written in a language that tells a computer what to do.



Speaking a computer's language ...

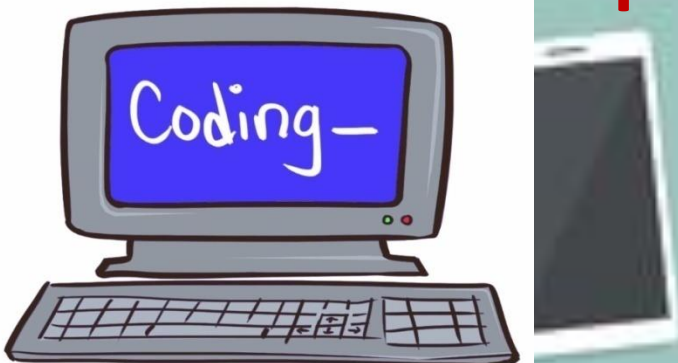


- **The most important thing is ...**
- **you can't see what's going on in the computer– you have to track the current 'state' of everything in your head.**
- **So, computer programs(code) have to be very precise because computers can't think for themselves**
- **Computers are great at following instructions(in the program) but computers will follow instructions even if they(programs) are wrong.**

Speaking a computer's language ...



- The best programs are often those which use the quickest possible way to complete a task.
- Therefore, to find the quickest possible way we used to represent Algorithms in many ways,
- Most commonly
 - flow charts and
 - pseudocodes.



Speaking a computer's language ...



- A programming language use to covert an algorithm to a coding.
- There are many different types of programming languages.
- Examples:
- Logo, Scratch, Blockly, Python and Kodu. (Each of these languages are suited to different things.)

Flow charts



- Algorithms and flowcharts are two different ways of presenting the process of solving a problem.
- Flowcharts are **diagrams** that visually present the process of solving problems.
- In flowcharts, those steps are usually displayed in shapes and process boxes with arrows.



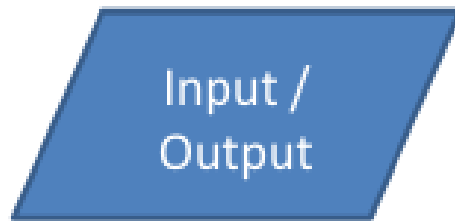
Flow chart Symbols



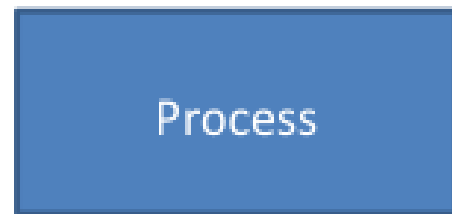
This marks the start or end of a process



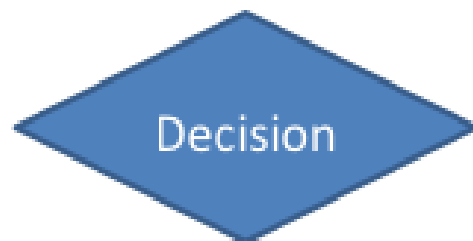
Denotes the direction of flow



Represents either an input or output operation



Denotes a process to be carried out



Represents a decision or branch. The result should follow one of two paths ("yes" or "no")

Pseudocodes



- Pseudocode allows you to 'get your ideas down' as step by step code-ish statements, without getting bogged down in the specifics of a particular language's syntax.
- Pseudocode uses the structure of a programming language, but is intended for humans, not computers, to read

Solving a problem



Step 1

- The problem needs to be analyzed well before solving it.
- Then you can get a good understanding of how to solve the problem.
- The process of problem solving has an input, an outcome and a process.
 - **Input** - Things to be included to solve the problem.
 - **Process** - Guidelines to be followed to solve the problem.
 - **Output** - The result you get after solving the problem.

Solving a problem ...



Example: Making a fruit salad

- The input, process and output of preparing a fruit salad is as follows.
 - **Input - a variety of fruits**
 - **Process - washing fruits, cutting fruit, mixing**
 - **Output - Fruit salad**

Solving a problem ...



Example: 2

- The input, process and output of finding the area of a rectangle are as follows.
 - **Input** - the length and the width of the rectangle
 - **Process** - length x width
 - **Output** - area of the rectangle

Solving a problem ...



**Step
2**

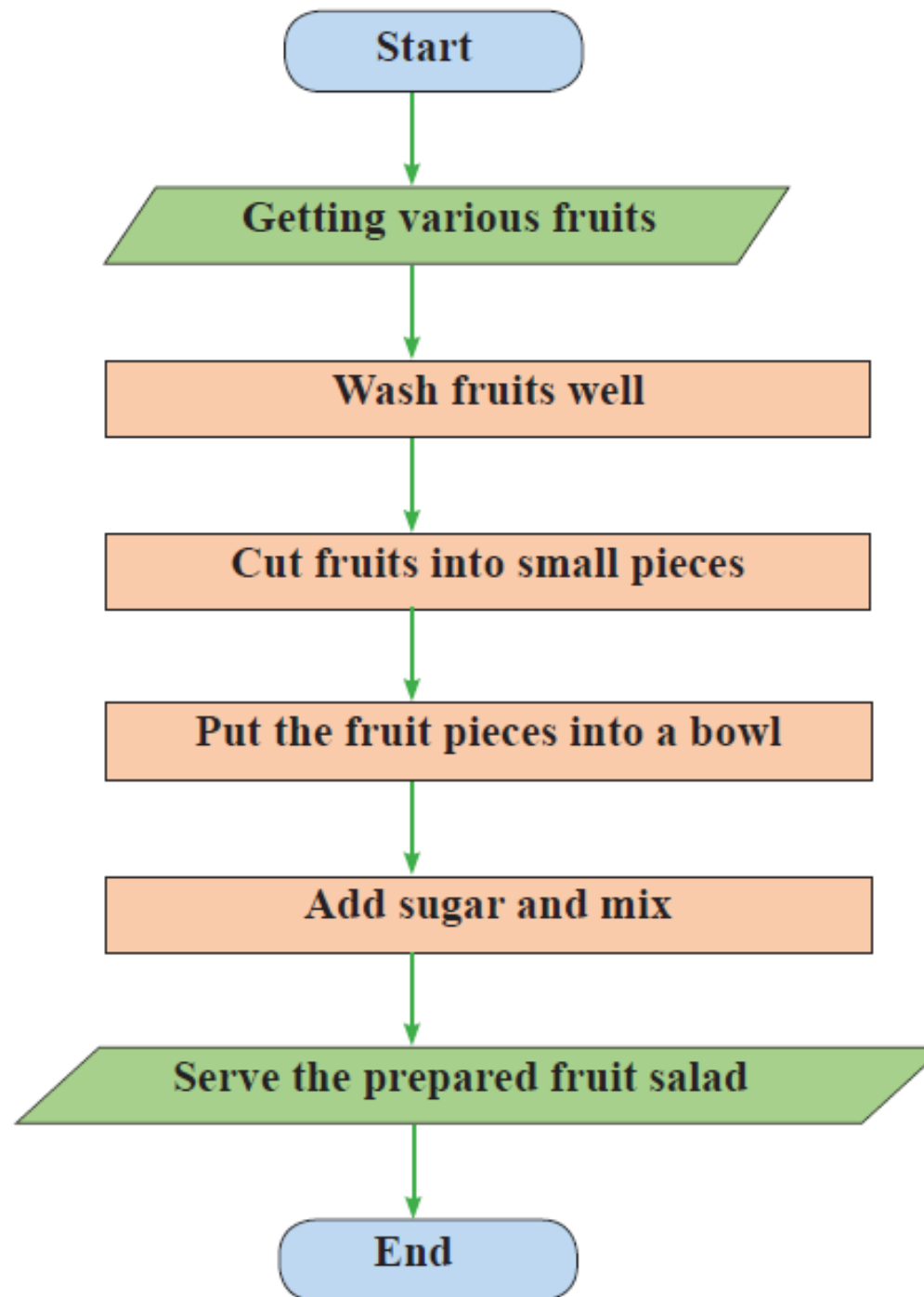
- Write algorithmic steps to solve the problem

Example

1. Start
2. Get various fruits
3. Wash fruits
4. Peel off fruits
5. Cut into pieces
6. Put into a bowl
7. Add some sugar
8. Mix together
9. Serve
10. End

- Using a pseudocode

- Using a Flow chart



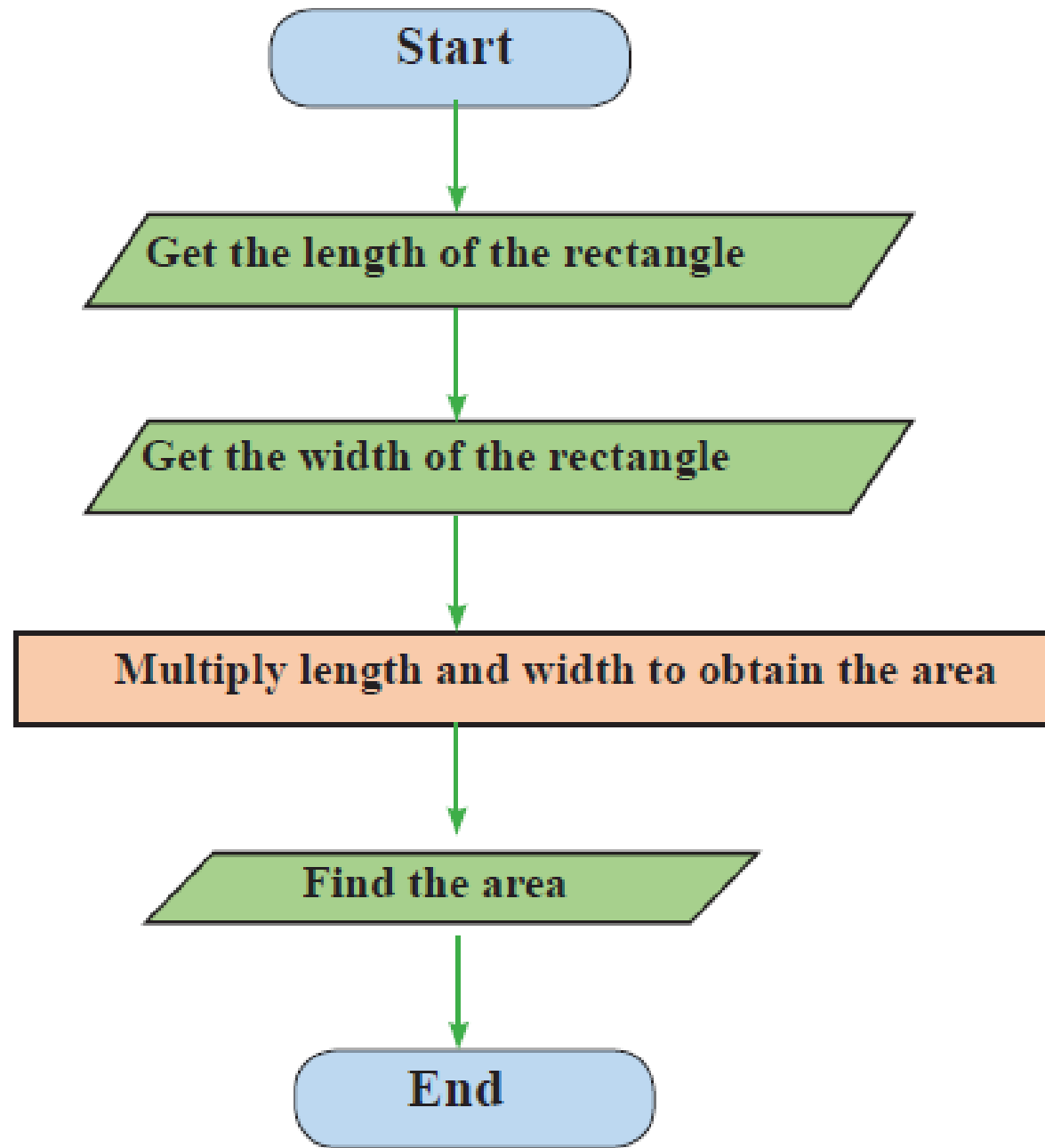
Solving a problem ...



Example

1. Start
2. Input length
3. Input width
4. Multiply length and width
5. Get the area of the rectangle
6. End

- Using a Flow chart



Solving a problem ...



**Step
3**

- **Convert the algorithmic steps into a programming language code**

Example :

- **Develop the program using Scratch coding blocks**

Home work



1. Write pseudocode steps and draw a flow chart for Adding two numbers
2. Write pseudocode steps and draw a flow chart for Calculating the Perimeter of the rectangle

