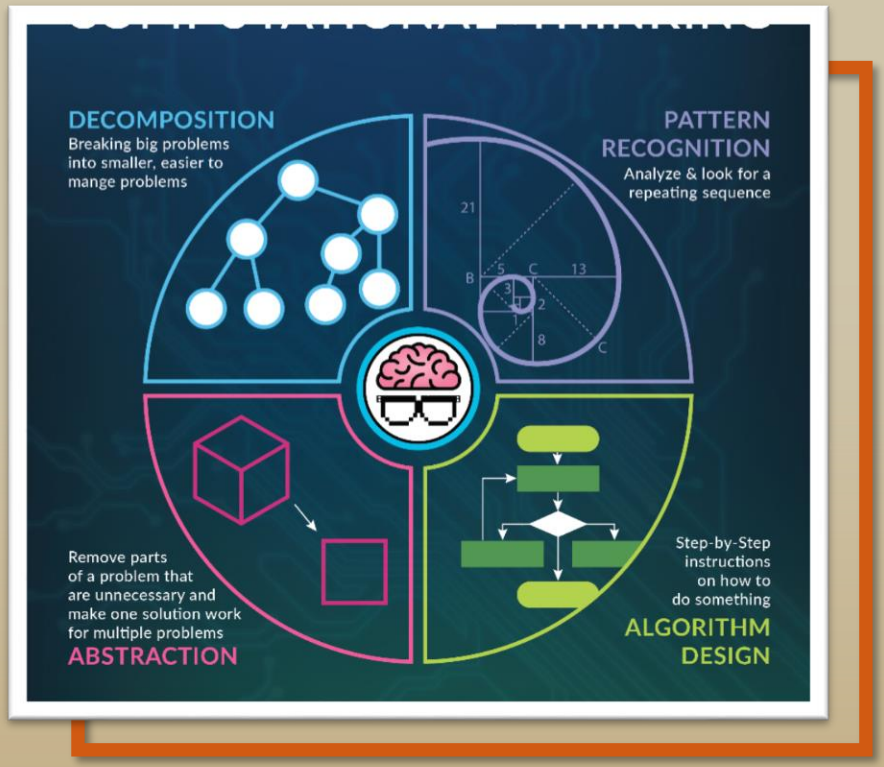
Computational Thinking



Picture credit: www.computationalthinkers.com/product/computationalthinking/

- Using special thinking patterns and processes to pose and solve problems or prepare programs for computations.
- Simply put, it is a set of skills that help to set up a problem in such a way that a computer can help you solve it.

Four pillars of Computational Thinking

> Breaking a problem down into smaller, more manageable parts.

Decomposition

> Finding similarities between items as a way of gaining extra information.

Pattern matching

> Ignoring certain details in order to come up with a solution that works for a more general problem.

Abstraction

> Controlling a process by automatic means, reducing human intervention to a minimum.

Algorithm (Automation)

Computational Thinking Resources

Video resources:
https://www.youtube.com/watch?v=kdngEhA4I00 (for younger children)
https://www.youtube.com/watch?v=qbnTZCj0ugI (for middle school children)
https://www.youtube.com/watch?v=qpxLusH4quY&list=PLiTx3ehKDhsdqt3F1tR_FSyHwXpdlp2ox&index=1 (focus on algorithm)

ISTE Toolkit:
https://www.iste.org/explore/Solutions/Computational-thinking-for-all

Code.org CS unplugged classroom activity ready to be used:
https://code.org/curriculum/course3/1/Teacher#Vocab

	3	4	
4			2
1			3
	2	1	

Activity 1.

Solve Sudoku

Decomposition

	3	4	
4			2
1			3
	2	1	

Work on upper-left region (1)

Work on upper-right region (2)

Work on lower-left region (3)

Work on lower-right region (4)

For Region 1

	3	4	
4			2
1			3
	2	1	

Step 1.

Look at the numbers that are missing in row 1 of the puzzle.

That would be 1, 2.

Step 2.

Look at the numbers that are missing in column 1 of the puzzle.

That would be 2 and 3.

Step 3.

Look at the numbers missing in region 1.

That would be 1, 2.

Step 4.

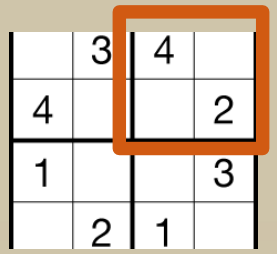
If there is only one number missing from all three sets, that number goes to row 1 and column 1 in region 1.

That would be 2.

Step 5.

If there is a second number missing from all three sets, continue to the next cell and come back when you have more information.

For Region 2



Look at the numbers that are missing in column 3 of the puzzle

That would be 2 and 3.

If there is only one number missing from all three sets, that number goes to row 2 and column 3 regaion 2.

That would be 3.

Step 1.

Look at the numbers that are missing in row 2 of the puzzle.

That would be 1, 3.

Step 2.

Step 3.

Look at the numbers missing in region 2.

That would be 1, 3.

Step 4.

Step 5.

If there is a second number missing from all three sets, continue to the next cell and come back when you have more information.

Pattern Matching

Step 1.

Look at the numbers that are missing in row 1.

That would be 1, 2.

For region 1

Step 1.

Look at the numbers that are missing in row 2.

That would be 1, 3.

For region 2

Abstraction

	3	4	
4			2
1			3
	2	1	

Look at the numbers that are missing in column Y of the puzzle.

That would be ...

If there is only one number missing from all three sets, that number goes to row X and column Y in region N.

That would be …

Look at the numbers that are missing in row X in the puzzle.

That would be ...

Look at the numbers missing in region N.

That would be 1, 3.

If there is a second number missing from all three sets, continue to the next cell and come back when you have more information.

Automation (algorithm)

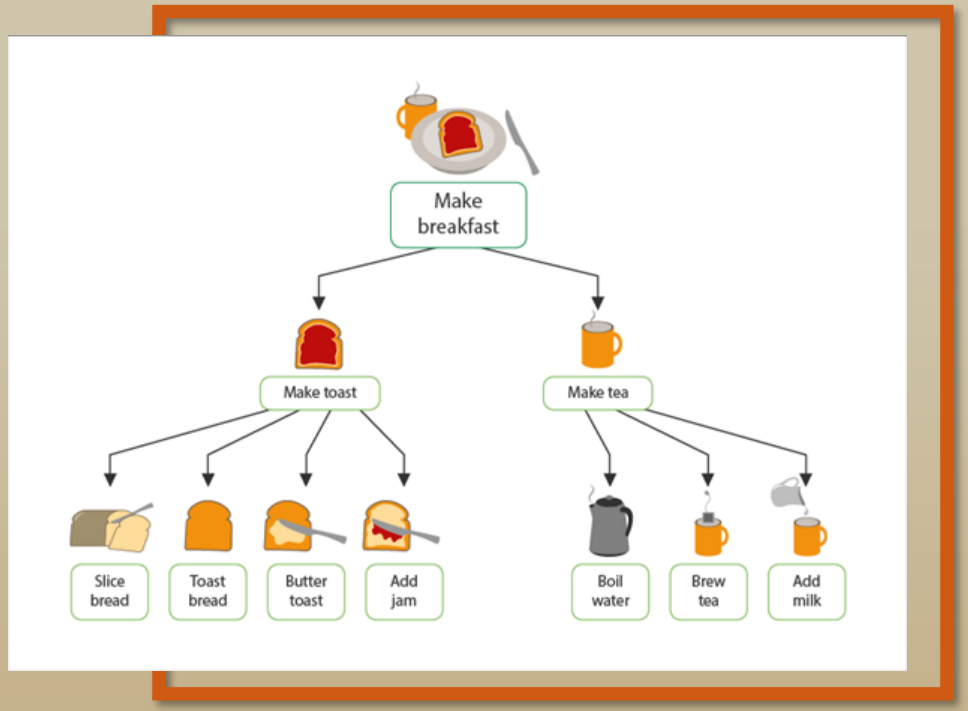
For every region in this puzzle

For every cell in a region

Decomposition

“Journey of a thousand miles begins with a single step.”
Lao Tzu

“When eating an elephant, take one bite at a time”
Creighton Abrams



Decomposition

In computing, decomposition is the process of breaking down a task into smaller, more-manageable parts.

Decomposition Resources

Tut, Clap, Jive at barefoot website

Divide and Conquer

Go to http://barefootcas.org.uk to create an account

https://sites.google.com/isabc.ca/computationalthinking/decomposition

Pattern Recognition (Matching)

Patterns are everywhere. By identifying patterns, we can create rules and solve more-general problems.

Pattern Resources

Cut Block Logic Puzzles by Queen Mary University of London (https://goo.gl/1ZUu5S)

https://sites.google.com/isabc.ca/computationalthinking/pattern-recognition

https://www.barefootcomputing.org/concepts-and-approaches/patterns

Abstraction

Caterpillar class Timetable	Monday	Tuesday	Wednesday	Thursday	Friday
8:50 to 9:00	Registration	Registration	Registration	Registration	Registration
9:00 to 10:00	English	English	English	English	English
10:00 to 10:20	Playtime	Playtime	Playtime	Playtime	Playtime
10:20 to 10:30	Class time	Class time	Class time	Class time	Class time
10:30 to 11:30	Maths	Maths	Maths	Maths	Maths
11:30 to 12:00	Phonics	Phonics	Phonics	Phonics	Phonics
12:00 to 1:00	Lunchtime	Lunchtime	Lunchtime	Lunchtime	Lunchtime
1:00 to 2:15	Topic	PE (small hall)	PE (large hall)	PPA subjects	Topic
2:15 to 3:15	Topic	Singing Assembly	Topic	PPA subjects	School Assembly

- The practice of ignoring certain details in order to come up with a solution that works for a more general problem.
- It is a way of letting go of details to make a process easier.

Abstraction Resources

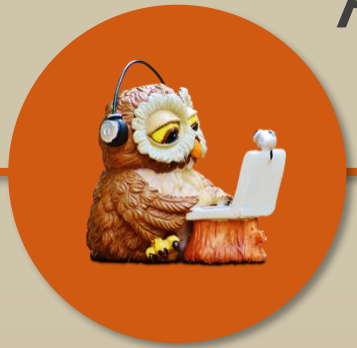
Code.org
https://studio.code.org/s/20-hour/stage/14/puzzle/1

Red Black Mind Meld: https://goo.gl/19QSzO

https://sites.google.com/isabc.ca/computationalthinking/abstraction

https://code.org/curriculum/course4/5/Teacher

Algorithm (Automation)



A process controlled by automatic means and reduce human intervention to a minimum.

Watch this video:
https://www.youtube.com/watch?v=qpxLusH4quY

Algorithm (Automation)

- An algorithm is a sequence of instructions or a set of rules to get something done.
- Algorithms are written for a human, rather than for a computer to understand. In this way, algorithms differ from programs.

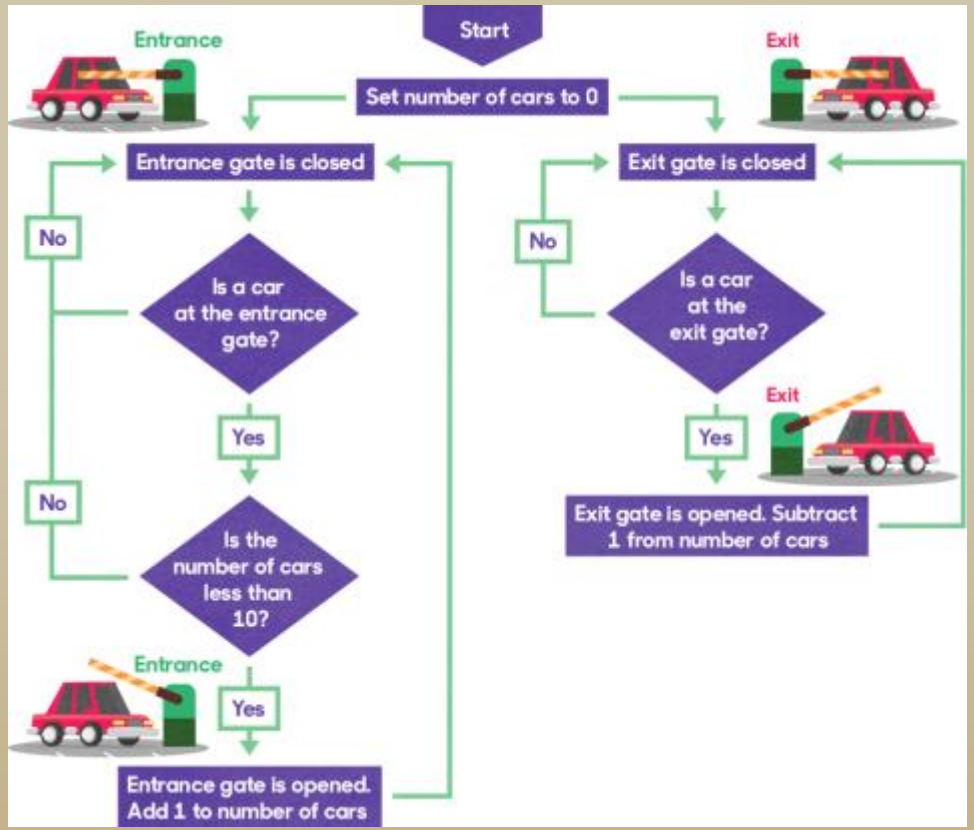
Algorithm example

Multiply by 10 and the number will move one place to the left

Thousands	Hundreds	Tens	Units
		1	5

Thousands	Hundreds	Tens	Units
	1	5	0

Flowchart


Start
Set number of cars to 0
Entrance
Exit
Entrance gate is closed
Exit gate is closed
No
Is a car at the entrance gate?
Yes
No
Is the number of cars less than 10?
Entrance
Yes
Entrance gate is opened. Add 1 to number of cars
No
Is a car at the exit gate?
Exit
Yes
Exit gate is opened. Subtract 1 from number of cars

Algorithm Resources

Divide and Conquer
https://docs.google.com/document/d/1DMDuomVc5gZ_NSaC3afERx4OuESWweydMICqetin3to/edit

Calculate surface area
https://docs.google.com/document/d/1pDf7DHtGvmFkrPxg0EmY0HAMoQsCoLaYsu9ZQWmba7g/edit

Sorting algorithm
https://classic.csunplugged.org/sorting-algorithms/

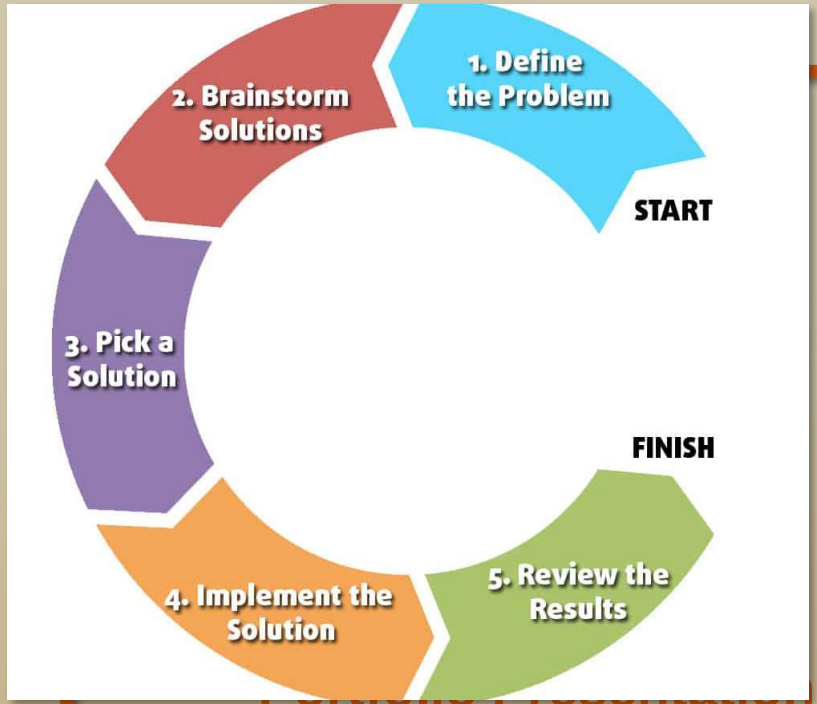
https://sites.google.com/isabc.ca/computationalthinking/algorithm

https://code.org/curriculum/course4/5/Teacher

Computational Thinking

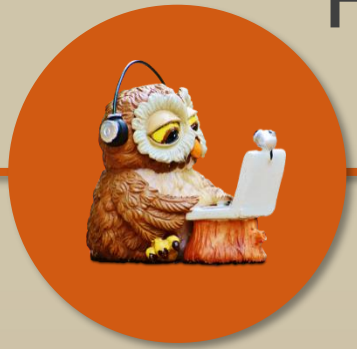
Unplugged Lesson in Action - Binary Bracelets from code.org

Problem-solving Process



- Problem analysis
- Alternative consideration
- Choosing an approach
- Problem decomposition
- Algorithm development
- Algorithm correctness
- Algorithm efficiency
- Reflection

Problem Understanding



Is the step that leads to the identification of the problem characteristics.

For algorithmic problems, it starts by recognizing the input categories of the problem and the selection of the required output category for each input category.

Basic strategies

Identify outputs

02

Identify inputs

01

If yes, how;
If not, why?

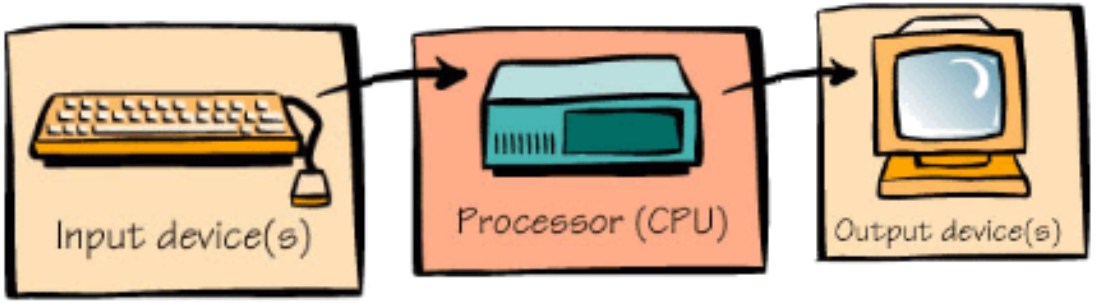
03

Each input is a subtask

04

Extreme cases

What Computers Do



Receive input

Process Information

Produce Output

What are inputs?

- Input is data sent to a computer system from devices such as a keyboard, mouse, microphone, camera or physical sensor.
- Input devices enable information from the outside world to get into a computer – without them, we wouldn't even be able to switch the computer on!
- Some are built in; others are plugged in or wirelessly connected.

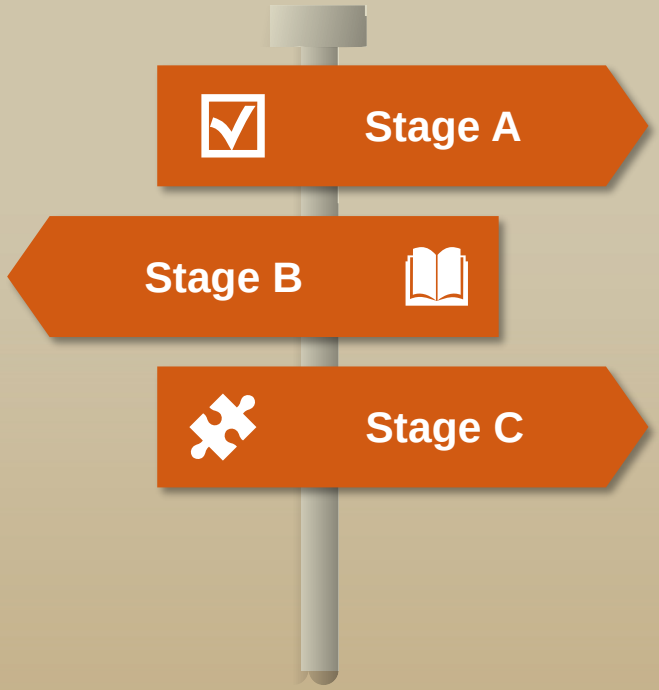
What are outputs?

Output is data or information communicated from a computer system to the outside world via various devices which include:

Activity 6 (a): Input/ Output Examination

Read the two problem specifications given by your instructor.

For each of them, select a technique learnt in Stage A to identify input/output for each problem.



Read the following article:

https://codeburst.io/10-steps-to-solving-a-programming-problem-8a32d1e96d74

Presentation and discussion

Solution Design

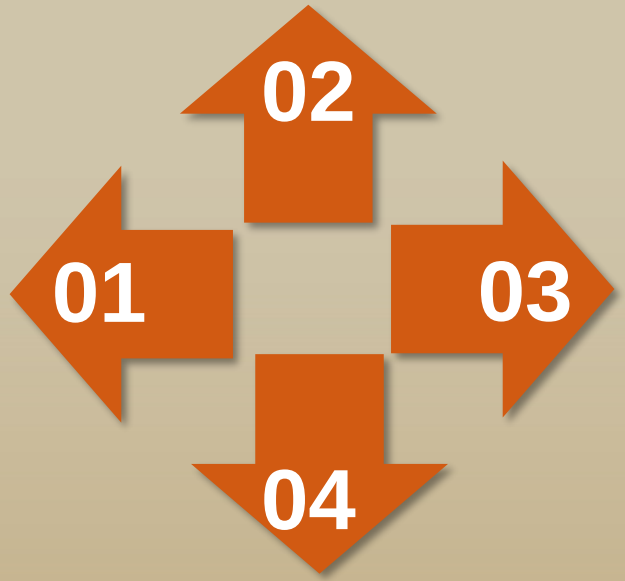
Is the major challenge facing novice learners.

The examination of a given problem’s inputs and outputs clarifies the problem to the problem solver. The next stage is to define the variables needed to solve the problem.

Basic Strategies

Stepwise refinement

Define problem variables



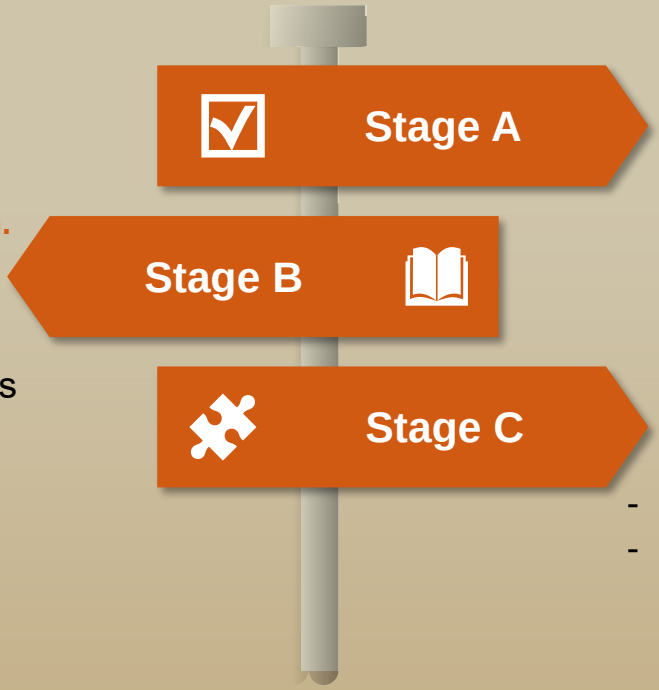
Identify algorithmic patterns

How should K-8 incorporate computer science into their curriculum?

Variables

A variable can be thought of as a box that can hold one value at a time.

Activity 6 (b): Choose problem variables



Further analyze the problems given in Activity 6(a).

Identify variables needed to solve them.

- Come up with variables needed
- Come up with guidelines for a teacher to evaluate each learner‘s investigation of variables

Presentation and discussion

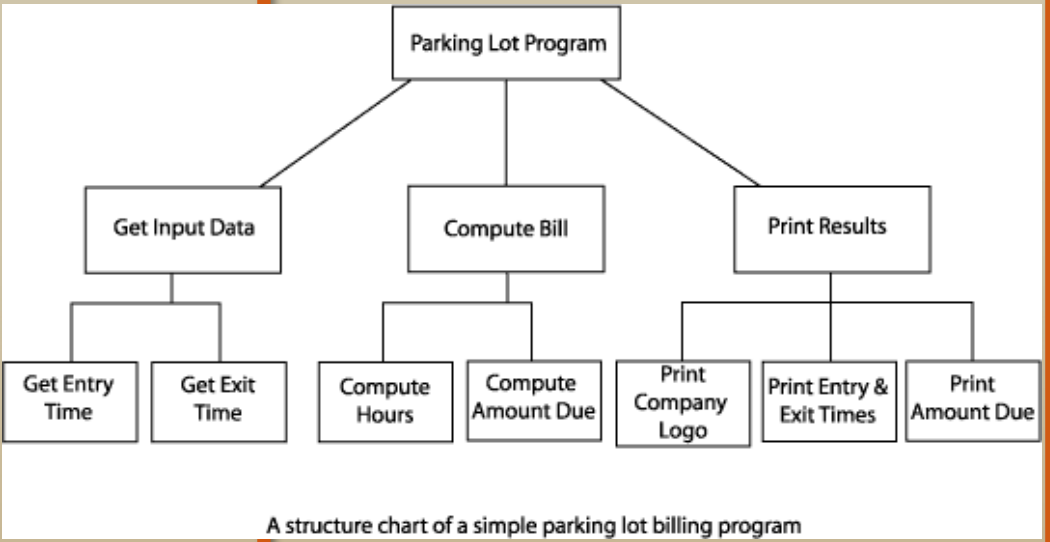
- Compare solutions
- Summarize guidelines for a teacher to evaluate each learner‘s investigation of variables

Stepwise Refinement

- Is a top-down methodology that first obtains the overview of the structure of the problem and the relationship among each parts, and then to address specific and complex issues related to the implementation of subpart.

- refers to the progressive refinement in small steps of a program specification into a program.

Step-wise (top-down) Refinement



A structure chart of a simple parking lot billing program

- was used first in the paper titled *Program Development by Stepwise Refinement* by Niklaus Wirth, the author of the programming language Pascal and other major contributions to software design and software engineering, in the *Communications of the ACM,* Vol. 14 (4), 1971, pp. 221-227.
- Wirth said: "It is here considered as a sequence of design decisions concerning the decomposition of tasks into subtasks and of data into data structures."

Stepwise

- Start with the initial problem statement
- Break it into a few general steps
- Take each "step", and break it further into more detailed steps
- Keep repeating the process on each "step", until you get a breakdown that is pretty specific, and can be written more or less in pseudocode
- Translate the pseudocode into real code

Stepwise Refinement (Example)

Problem Statement:
Determine the class average for a set of test grades, input by the user. The number of test grades is not known in advance (so the user will have to enter a special code -- a "sentinel" value -- to indicate that he/she is finished typing in grades).

Initial breakdown into steps

- Declare and initialize variables
- Input grades (prompt user and allow input)
- Compute class average and output result

Now, breaking down the "compute" step further, we got:

Compute:

- ✓ add the grades
- ✓ count the grades
- ✓ divide the sum by the count

Revised breakdown of steps

- Declare and initialize variables
- Input grades -- count and add them as they are input
- Compute class average

Breaking the steps into smaller steps

So, now we can break down these 3 steps into more detail. The input step can roughly break down this way:

loop until the user enters the sentinel value (-1 would be good)

- prompt user to enter a grade (give them needed info, like -1 to quit)
- allow user to type in a grade (store in a variable)
- add the grade into a variable used for storing the sum
- add 1 to a counter (to track how many grades)

Breaking the steps into smaller steps

We could specifically write this as a while loop or as a do-while loop. So one more refining step would be a good idea, to formulate the pseudo-code more like the actual code we would need to write. For example:

do

- prompt user to enter a grade (give them needed info, like -1 to quit)
- allow user to type in a grade (store in a variable)
- add the grade into a variable used for storing the sum
- add 1 to a counter (to track how many grades)

while user has NOT entered the sentinel value (-1 would be good)

Breaking the steps into smaller steps

If we look at this format, we realize that the "adding" and "counting" steps should only be done if the user entry is a grade, and NOT when it's the sentinel value. So we can add one more refinement:

do

- prompt user to enter a grade (give them needed info, like -1 to quit)
- allow user to type in a grade (store in a variable)
- if the entered value is a GRADE (not the sentinel value)
    - add the grade into a variable used for storing the sum
    - add 1 to a counter (to track how many grades)

while user has NOT entered the sentinel value (-1 would be good)

Breaking the steps into smaller steps

This breakdown helps us see what variables are needed, so the declare and initialize variables step can be now made more specific:

initialize variables:

- a grade variable (to store user entry)
- a sum variable (initialized to 0)
- a counter (initialized to 0)

Compute and print:

- divide sum by counter and store result
- print result

Putting it all together

initialize variables:
  a grade variable (to store user entry)
  a sum variable (initialized to 0)
  a counter (initialized to 0)

grade entry:
  do
    prompt user to enter a grade (give them needed info, like -1 to quit)
    allow user to type in a grade (store in a variable)
    if the entered value is a GRADE (not the sentinel value)
      add the grade into a variable used for storing the sum
      add 1 to a counter (to track how many grades)
  while user has NOT entered the sentinel value (-1 would be good)

Compute average:
  divide the sum by the counter
  print the answer

Variables

Insert the title of your subtitle Here

Variables are memory location used to keep information!

- Variables are used as placeholders for values such as numbers or words.
- Variables allow for a lot of freedom in programming.
- Instead of having to type out a phrase many times or remember an obscure number, computer scientists can use variables to reference them.

Variable Resource

https://code.org/curriculum/unplugged

https://www.barefootcomputing.org/concepts-and-approaches/variables

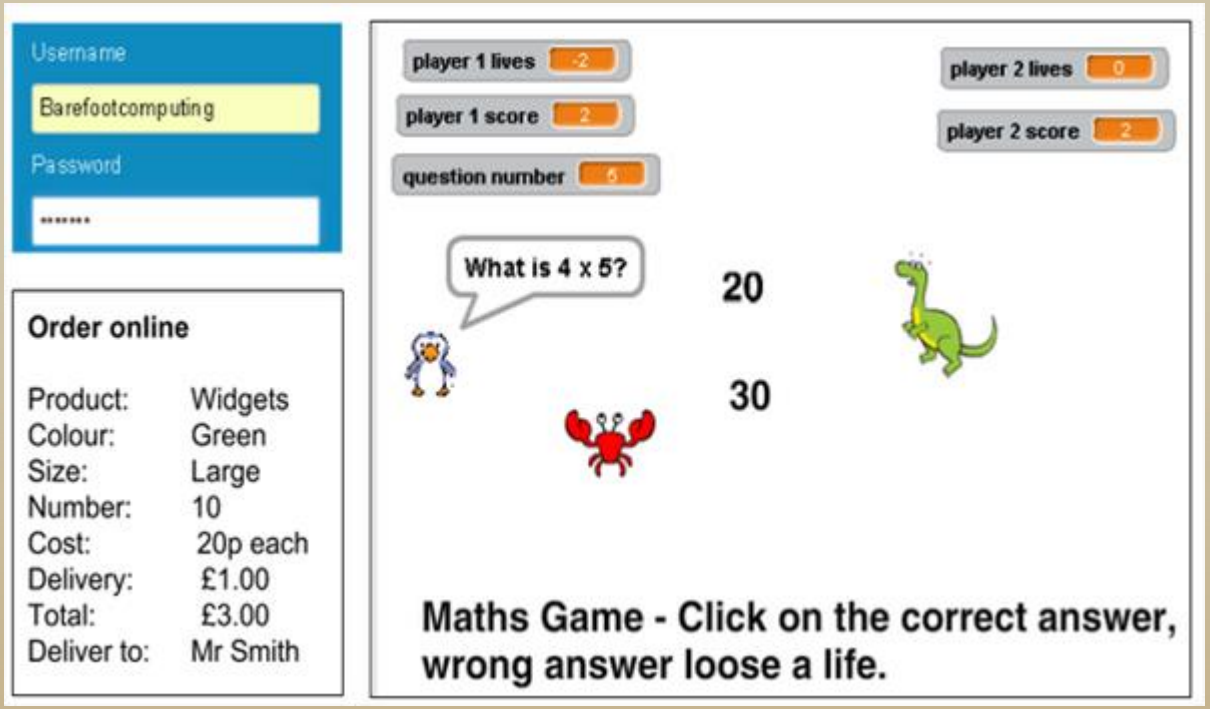
Code.org variable demo (1)

Code.org variable demo (2)

What is variable?

- A variable is a simple way of storing one piece of information somewhere in the computer’s memory while a program is running, and of getting that information back later.
- A variable is an example of a data structure.
- A variable can be numerical, textual or perhaps an indicator of true/false.
- Programs can store, retrieve or change the values of variables.

Variables used in a graphic user interface



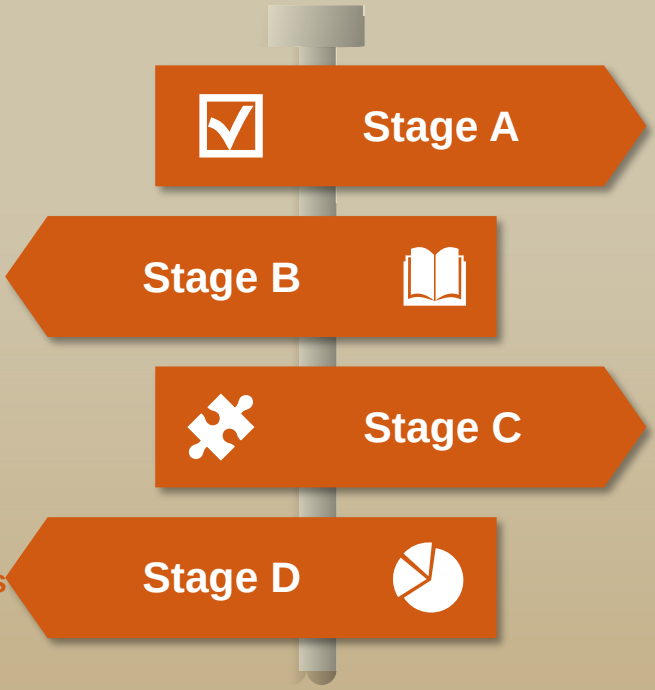
Variables

A variable can be thought of as a box that can hold one value at a time.

Declaring a variable

Before you can use a variable, you need to claim a piece of memory and associate it with a name.

Activity 1: Variables Unplugged Activity (peer teaching)



Download lesson plan at
https://www.barefootcomputing.org/resources/variables-unplugged-activity
or
https://www.barefootcomputing.org/resources/scratch-maths-quiz-variables

Prepare ppt

Teach the lesson

Discussion and reflections

Activity 2: Kidbots! (peer teaching)

Stage A

Select one from the 1,2,3,4

https://csunplugged.org/en/topics/kidbots/unit-plan/

Prepare ppt and other materials

Stage B

Stage C

Teach the lesson

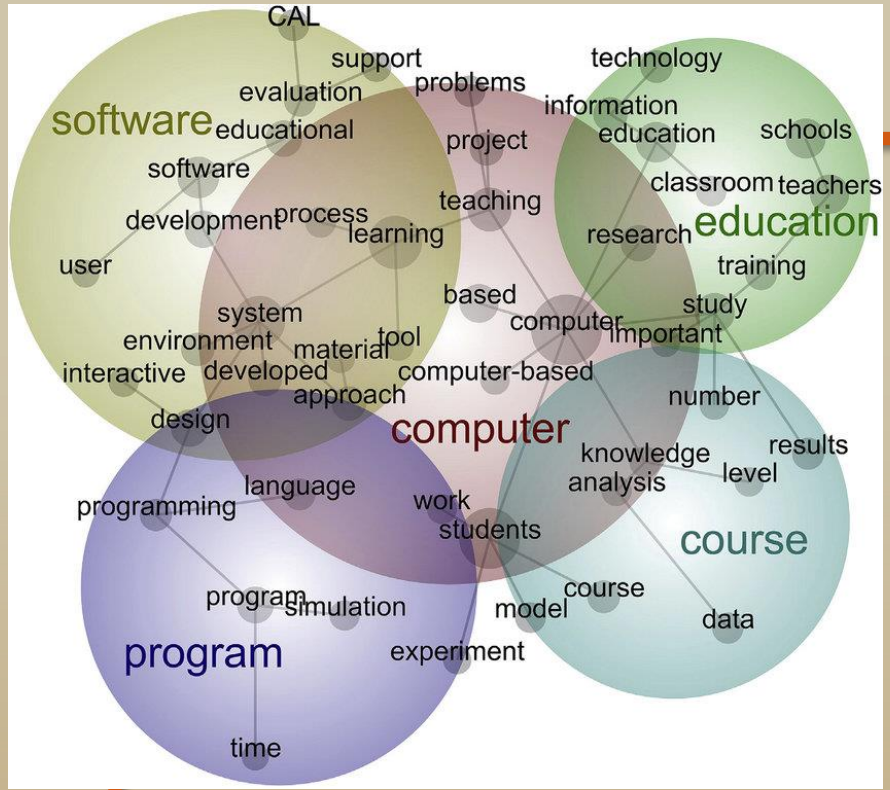
Design your own activity

Stage D

Step E.

Discussion and reflections

Cons and pros



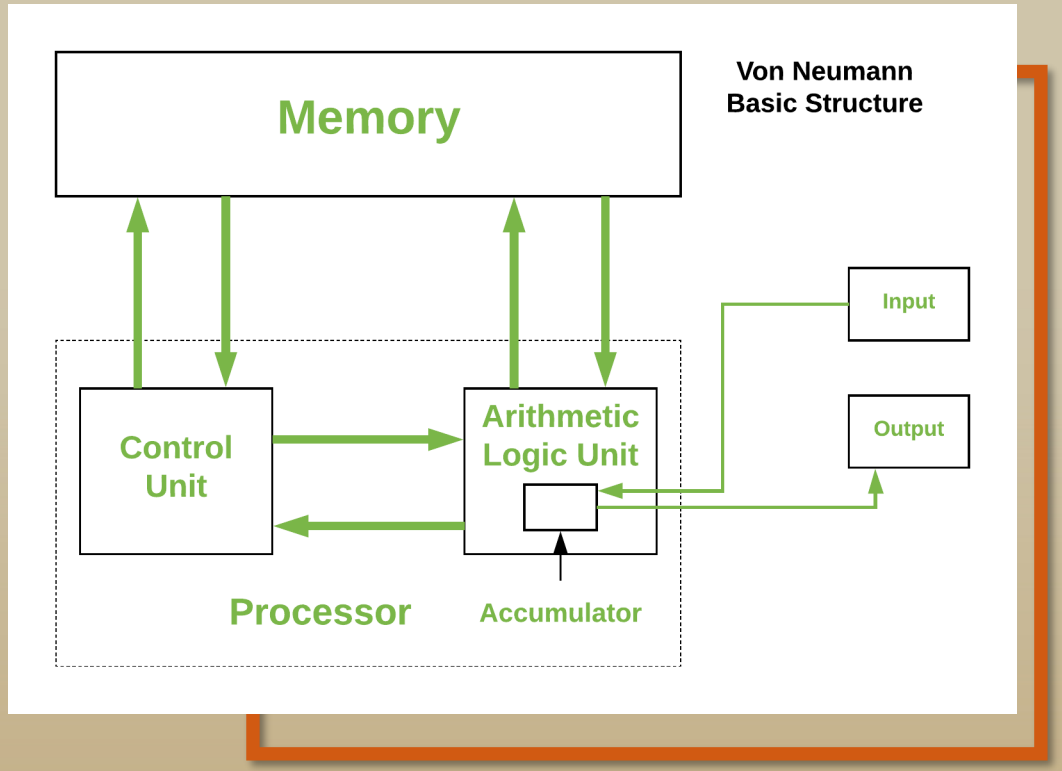
Concept maps

is a two-dimensional, graphic or schematic diagram illustrating the interconnections, and often the hierarchy, of a particular concept or topic.

How to create concept map? (1)

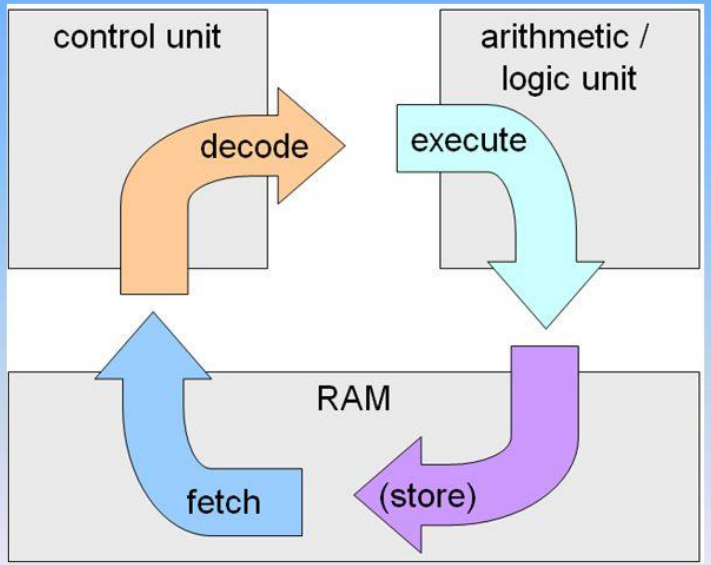
How to create concept map2 (2)

Von Neumann Computer Architecture



- **Fetch** – gets the next program command from the computer's memory
- **Decode** – deciphers what the program is telling the computer to do
- **Execute** – carries out the requested action
- **Store** – saves the results to a Register or Memory

Simple Layout of Fetch-Execute Cycle



Fetch – gets the next program command from the computer's memory
Decode – deciphers what the program is telling the computer to do
Execute – carries out the requested action
Store – saves the results to a Register or Memory

Activity 3: Pedagogical Examination of Concept Maps

Stage A

Read the article below:

https://www.nsta.org/publications/news/story.aspx?id=53174

Stage B

Design a classroom activity that ask students to create a concept map on a selected subject and create evaluation rules

Stage C

Present your class room activity and ask your peers to complete the classroom activity

Stage D

Present the concept maps created and evaluate them

Main concepts should be included and the relationship should be specified

Step E.

Discussion and reflections

-how can concept map help to promote learning

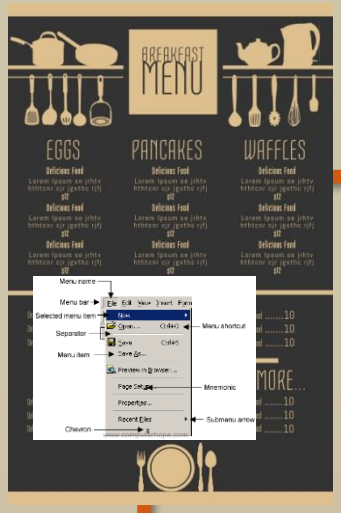
- How CM help assessment of learning outcomes

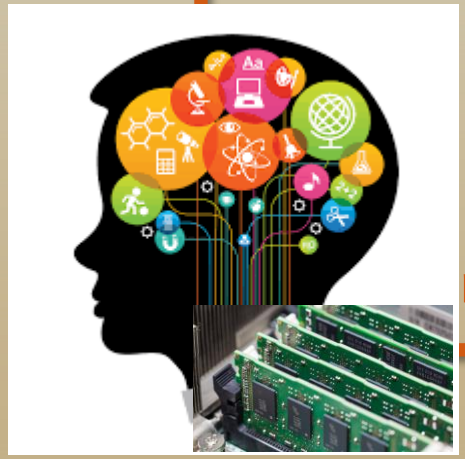
Concept Mapping Resource

https://ctl.byu.edu/tip/concept-mapping

http://www.inspiration.com/visual-learning/concept-mapping

https://www.nsta.org/publications/news/story.aspx?id=53174





Metaphors

Are used in order to understand and experience one specific thing by using an analogy to another thing, usually a familiar concept.

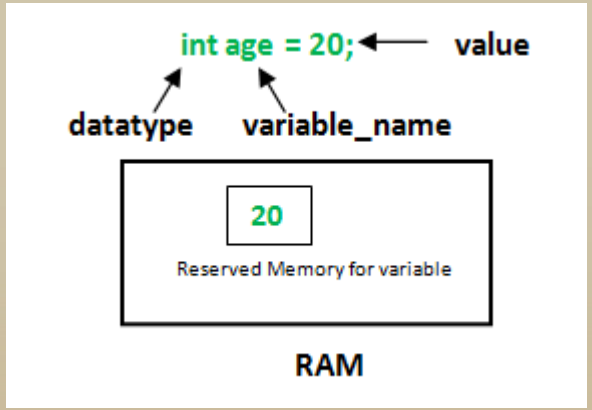
A variable can be thought of as a container that can hold one value at a time.

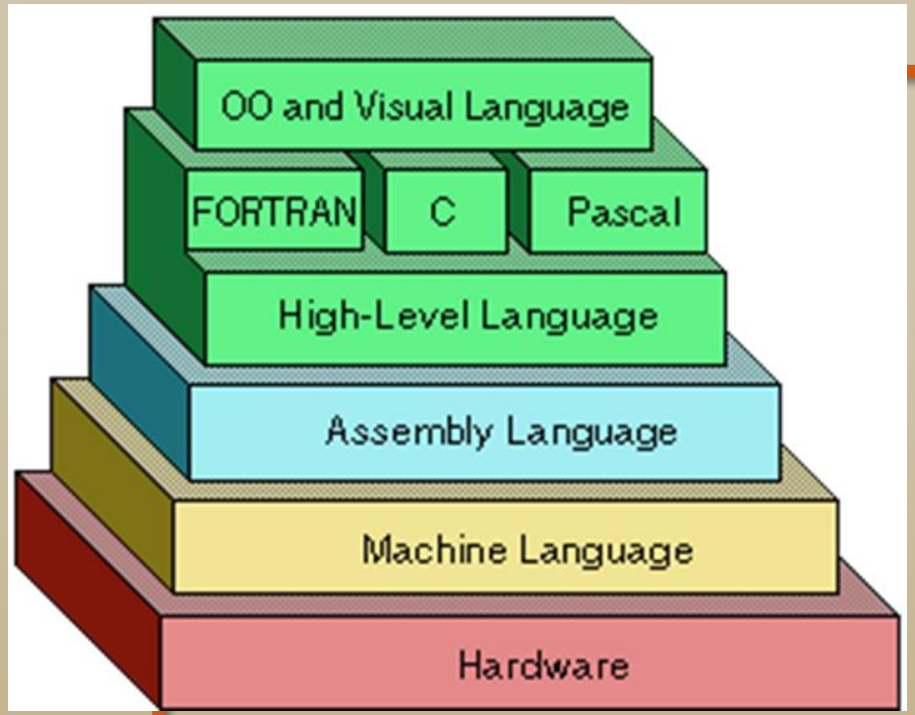
Data Types

- A variable in Java is designed to hold only one particular type of data.
- There are eight primitive types built into Java.
- **byte, short, int, long** variables hold integers.
- **float, double** variables hold real numbers
- **char** variable holds a single character from the Unicode character set.
- **boolean** variable holds one of the two logical values true or false.

Variable Declaration

```
int age = 20;

byte nextInStream;

short hour;

long totalNumberOfStars;

float reactionTime;

double itemPrice;
```





Classification

is used to classify objects and phenomena from real life to support and guide the mental construction of computer science concepts.

Declare problem variables

Mapping problem characters to different variables


YES / NO QUESTIONS
Am
Is
Are
Do
Does
Have
Has


George Smith
Carrie Smith
Chris Heinrich
George Smith
Caryn Smith
Lucricia Jones
Lasercrafting


False
True


CHARACTER


Average Calculator
Values (you may enter up to 30 numbers)
#1 1
#2 2
#3 3


int
float
boolean
string

Activity 4: Classification of Variable According to Types

Stage A

Study the pictures in previous page

Stage B

Sort them into different bins

Stage C

Define a variable for each scenario

You can simply impress your audience and add a unique zing.

Stage D

Design your own activity

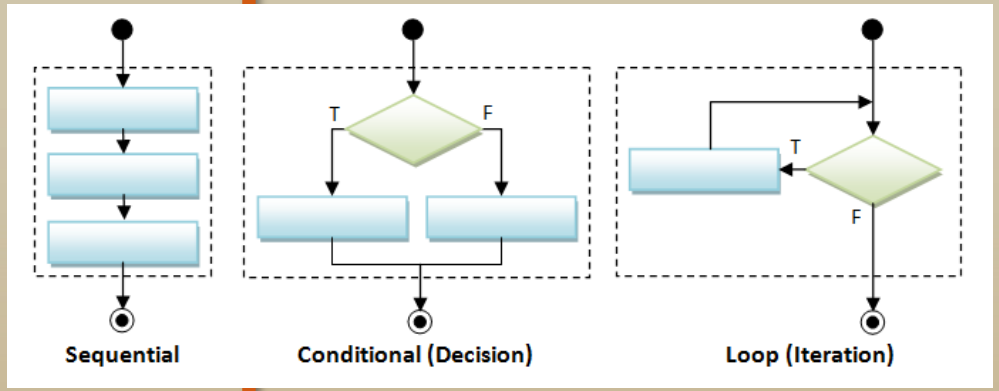
You can simply impress your audience and add a unique zing.

Step E.

Discussion and reflections

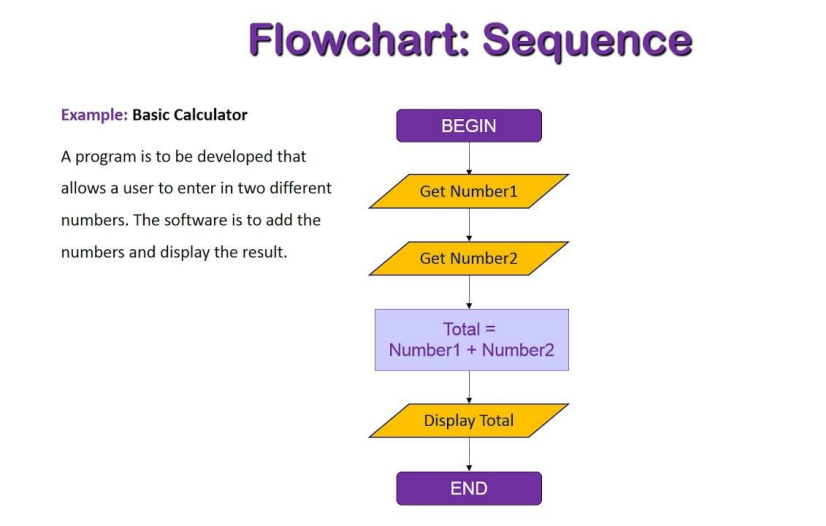
Cons and pros

Program Control Structure (using concept map)



Sequential: default mode. Sequential execution of code statements (one line after another) -- like following a recipe
Selection: used for decisions, branching -- choosing between 2 or more alternative paths.
Repetition: used for looping, i.e. repeating a piece of code multiple times in a row.

Sequence statements




```
int number1;
 int number2
 int total=0;

 Scanner input = new Scanner(System.in);

 System.out.println("please enter the first number");
 number1 = input.nextInt();

 System.out.println("please enter the second number");
 number2 = input.nextInt();
 total = number1+number2;

  System.out.println("your answer is" + total);
```