Problem Solving With Computational Thinking

Session-2


Kahoot!

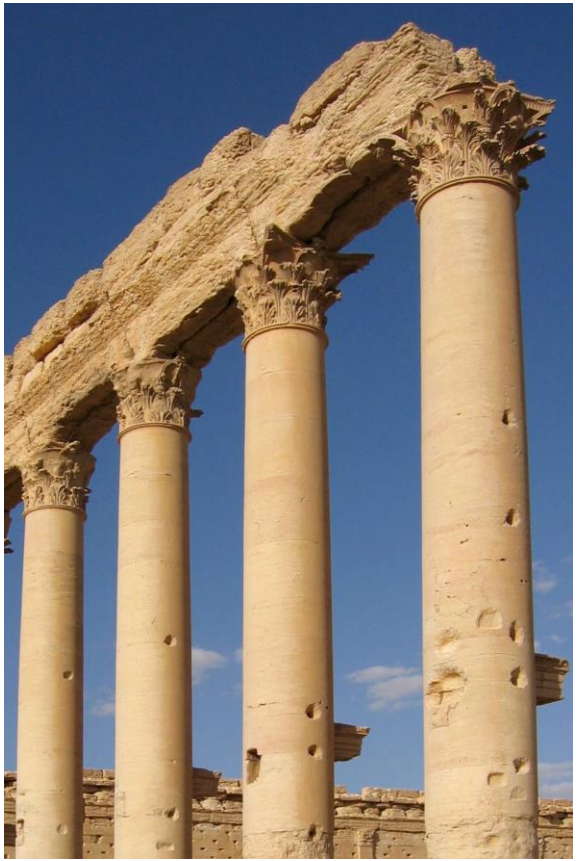
Review

- What is CT?

  Problem solving technique for humans to easen life

Review

- Decomposition
- Pattern Recognition
- Abstraction
- Algorithm Design

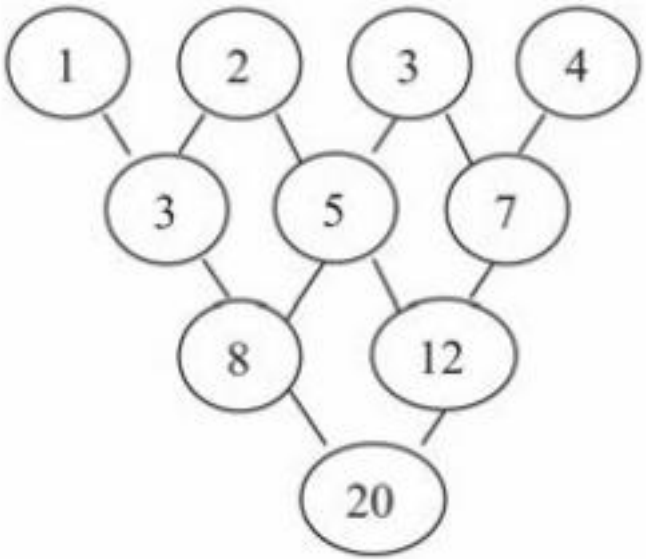


Review

Algorithm

- Step by step
- Clearly defined
- One simple job at a time
- Instruct computer what to do

Find the “?”



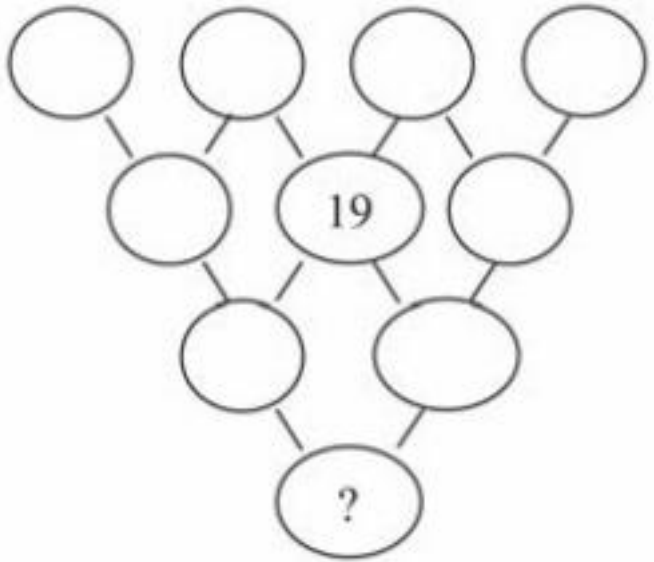
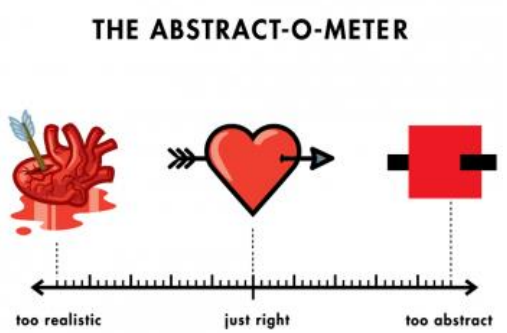


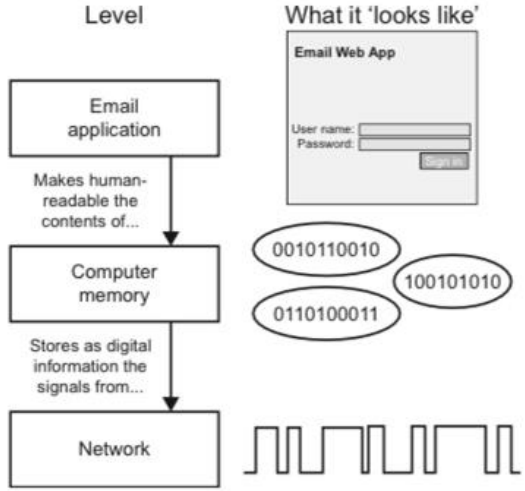
Table of Contents

Abstraction

Abstraction is getting rid of the useless information that is not going to have any contribution to the solution.

Abstraction is the core concept of computer science and computational thinking. To be able to express a real world problem to a computer, the problem has to be abstract.





Let's abstract stuff!

The key part of abstraction is ignoring the useless aspects of something and including the beneficial aspects. So, find out the useless aspects that are going to be ignored and the beneficial aspects that are going to be considered of the items below. Try going abstract as far as possible. (For example a house is nothing but a shelter for humans when you look abstractly)

Car 

Pencil 

House 

Mobile phone 

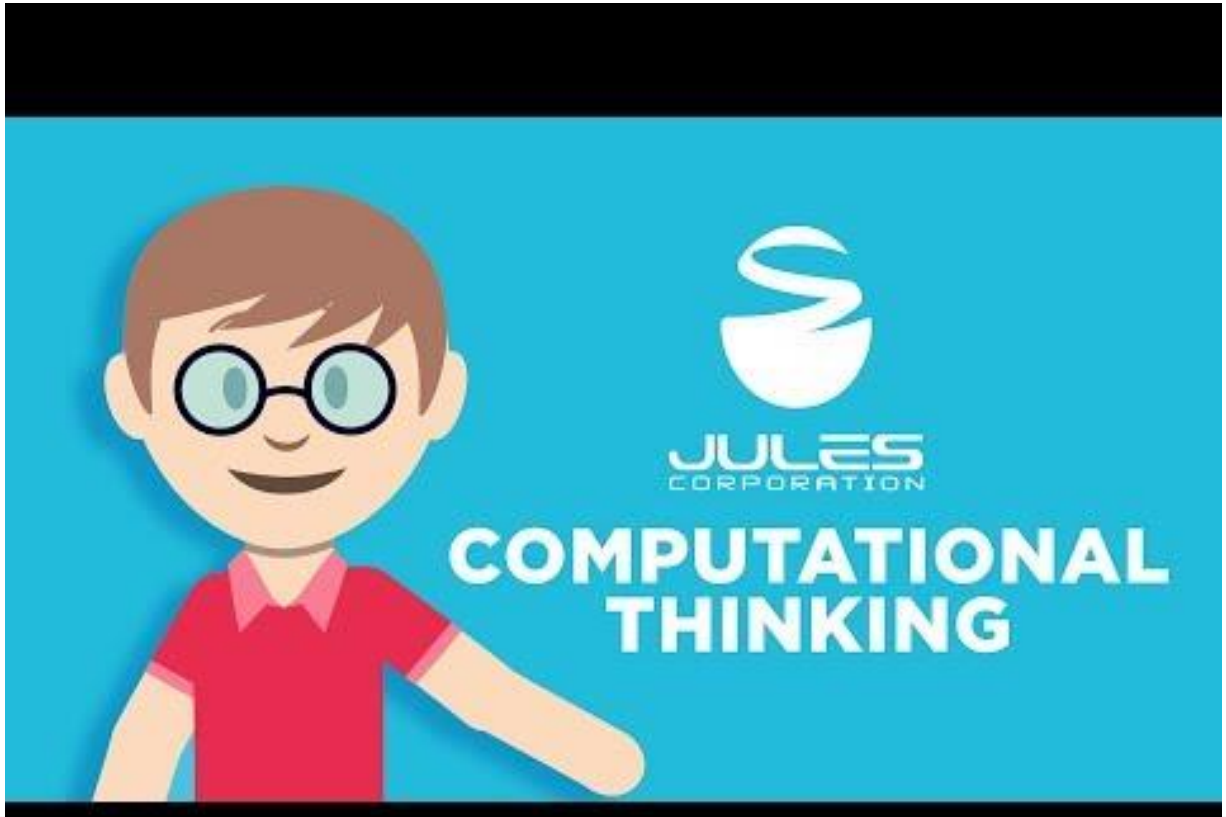
HINT: Think of what do these objects help solving and which aspects of them wouldn't prevent them from solving that particular problem.

Let's Abstract Stuff!

The abstractions should look something like this:

Object	Include	Ignore
Car	Engine, tires, rims, seat	Color, shape, trunk, radio
Pencil	color of the tip, grip	material, brand, type
Mobile phone	speaker, microphone,cellular	color, brand, camera

Recap Time



Students browse: www.youtube.com/embed/mUXo-S7gzds?cc_load_policy=1&cc_lang_pref=tr...

1 Pseudocode

Pseudocode

Let's discuss and try to predict what does pseudocode mean!

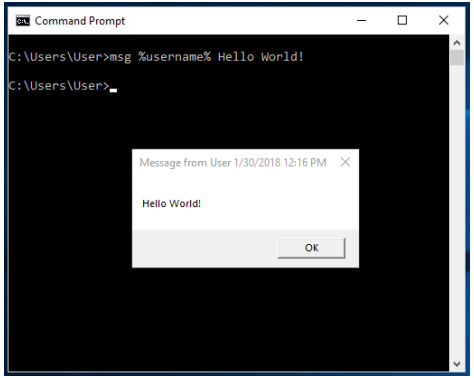
Pseudocode

- Pseudocodes are one of two popular ways to represent an algorithm.
- Pseudocode is an informal way of representing a computer program or an algorithm.
- It looks like a programming language though, it should be written in a programming language for it to be executed. It's language-agnostic.
- Writing pseudocode is basically writing what you want your programm to do in English.
- Aims to mimic the general style of a programming language

```
OUTPUT 'What is your name?'
INPUT user inputs their name
STORE the user's input in the name variable
OUTPUT 'Hello' + name
OUTPUT 'How old are you?'
INPUT user inputs their age
STORE the user's input in the age variable
IF age >= 70 THEN
    OUTPUT 'You are aged to perfection!'
ELSE
    OUTPUT 'You are a spring chicken!'
```

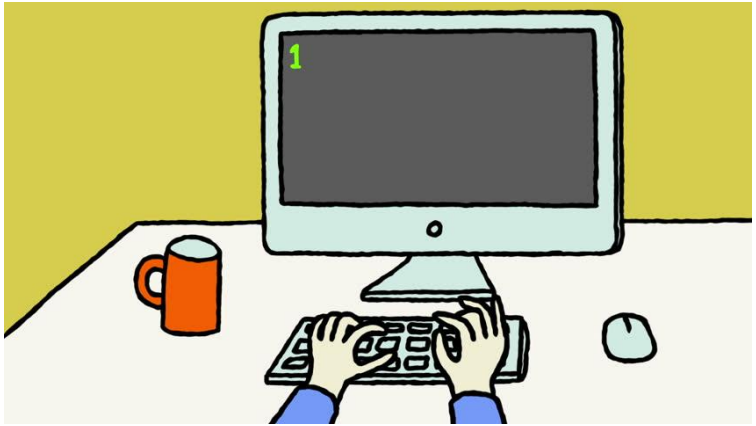
Pseudocode

```
OUTPUT 'What is your name?'
INPUT user inputs their name
STORE the user's input in the name variable
OUTPUT 'Hello' + name
OUTPUT 'How old are you?'
INPUT user inputs their age
STORE the user's input in the age variable
IF age >= 70 THEN
    OUTPUT 'You are aged to perfection!'
ELSE
    OUTPUT 'You are a spring chicken!'
```



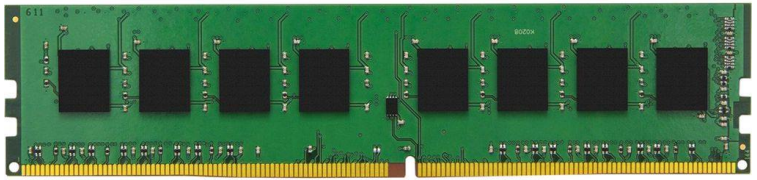
Pseudocode

```
OUTPUT 'What is your name?'
INPUT user inputs their name
STORE the user's input in the name variable
OUTPUT 'Hello' + name
OUTPUT 'How old are you?'
INPUT user inputs their age
STORE the user's input in the age variable
IF age >= 70 THEN
    OUTPUT 'You are aged to perfection!'
ELSE
    OUTPUT 'You are a spring chicken!'
```



Pseudocode

```
OUTPUT 'What is your name?'
INPUT user inputs their name
STORE the user's input in the name variable
OUTPUT 'Hello' + name
OUTPUT 'How old are you?'
INPUT user inputs their age
STORE the user's input in the age variable
IF age >= 70 THEN
    OUTPUT 'You are aged to perfection!'
ELSE
    OUTPUT 'You are a spring chicken!'
```



Keyword

There are these keywords that are widely used, you can use your own keywords, but these are the most frequently used amongst other computer programmers and should not be used as variable names.

```
START, BEGIN: This is the start of your pseudocode.
INPUT: This is data retrieved from the user through the input device.
READ, GET: This is used when reading data from a data file.
PRINT, DISPLAY, SHOW, OUTPUT: This will show your output to a screen.
COMPUTE, CALCULATE: To calculate the result of the expression.
SET, INIT: To initialize values
INCREMENT, BUMP: To increase the value of a variable
DECREMENT: To reduce the value of a variable
END: This is the end of your pseudocode
```

Example

Set total to zero

Set grade counter to one

While grade counter is less than or equal to ten

    Input the next grade

    Add the grade into the total

Set the class average to the total divided by ten

Print the class average.

Question

Let’s write a pseudocode for calculating Mary’s wage.

Inputs : hours and rate

Output: pay

Let's brew coffee

- Prepare ingredients
- Make coffee
- Prepare serving
- Enjoy

IF - ELSE IF - ELSE

This keyword is used if a certain condition has to be met for the upcoming block to be executed. For example:

```
IF you are happy
        Then smile
ENDIF
```

```
If you are tired
        Then rest
else if you are stressed
        Then relax
else
        Keep working
```

As you can see we also use indentation in order to declare that “smile” is being executed inside the if statement above it.

IF - ELSE IF - ELSE

This keyword is used if a certain condition has to be met for the upcoming block to be executed. For example:

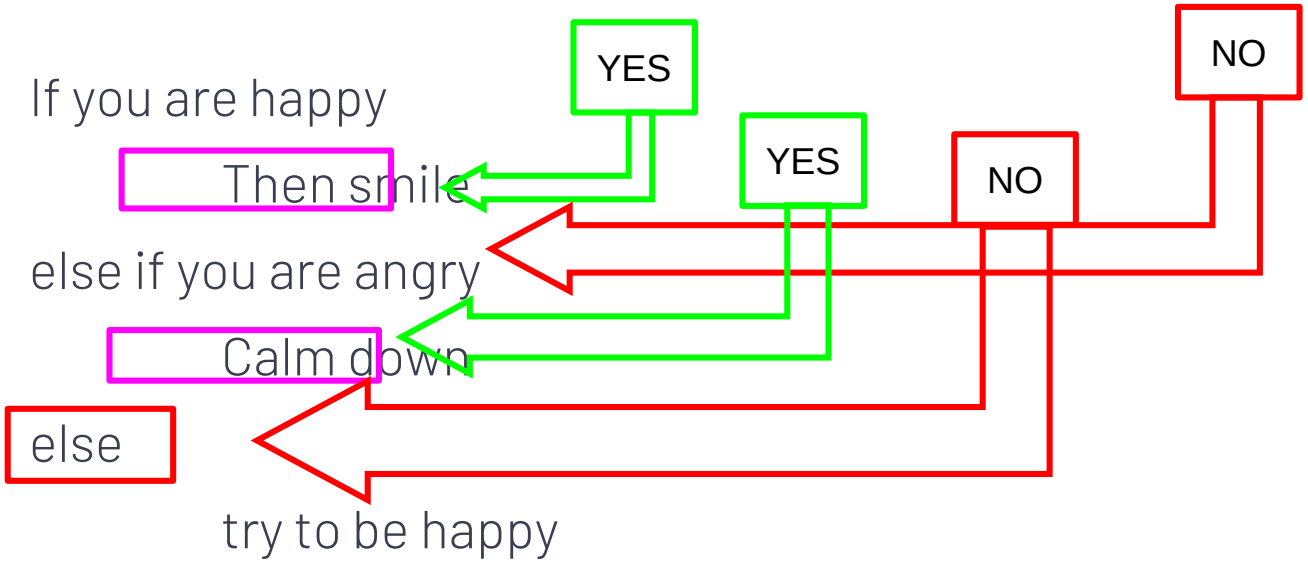
```
IF you are happy
        Then smile
ENDIF
```

```
If you are tired
        Then rest
else if you are stressed
        Then relax
else
        Keep working
```

As you can see we also use indentation in order to declare that “smile” is being executed inside the if statement above it.

IF - ELSE IF - ELSE

This keyword is used if a certain condition has to be met for the upcoming block to be executed. For example:



IF - ELSE IF - ELSE

```
Begin
INPUT hours, rate
IF hours < 40
THEN
    pay = hours * rate
ELSE
    pay = 40 * rate + (hours - 40) * rate *1.5
OUTPUT pay
End
```

FOR structure

For loop runs for each element inside a group. For example:

```
For every day of the week
        Count;
endfor
```

FOR structure

For loop runs for each element inside a group.
For example:

For every 25 minutes of study

Earn one Pomodoro;

endfor

Pomodoro = Pomodoro +1

WHILE Structure

While is similar to the for loop, differently it runs the loop until the condition provided is unsatisfied. Example:

```
Apples = 5
Oranges = 10
While apples < oranges
        increase apples;
endwhile
```

Exercise

Write a pseudocode that takes a number as an input and prints true if it is greater than 10 and false otherwise.

Let's wash the dishes

Let's wash the dishes. Think that we have all the tools etc.

Let's wash the dishes

```
gather the dirty dishes
    if you have a dishwasher around you
        put the dirty dishes inside the dishwasher
        set the settings of the dishwasher
        while the time set is not over
            wait
    else
        while dishes are not clean
            take one of the dishes
            wash it with your hand
            dry it and put it aside
```


Kahoot!

THANKS!

Any questions?

You can find me at:

- @marcus
- marcus@clarusway.com