

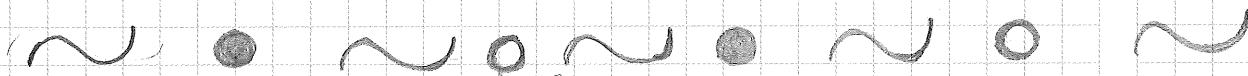
What is Computational Thinking?

Computational thinking is:

* Logical thinking (Good reasoning)

* Critical thinking

* Problem solving



Algorithm

① Decomposition: Breaking problems into parts

(Altından kalkabilcek küçük parçalara ayırmak)

sayılarla

③ Abstraction: Organising data/information logically

(Detayları alıp öğrenme, sonraya bırak.)

(Get rid of details)

② Pattern Recognition: Interpreting patterns and models

(Bir çözümde farklı değişikliklerle uyumlayabilme)

→ is series of instructions to carry out a task

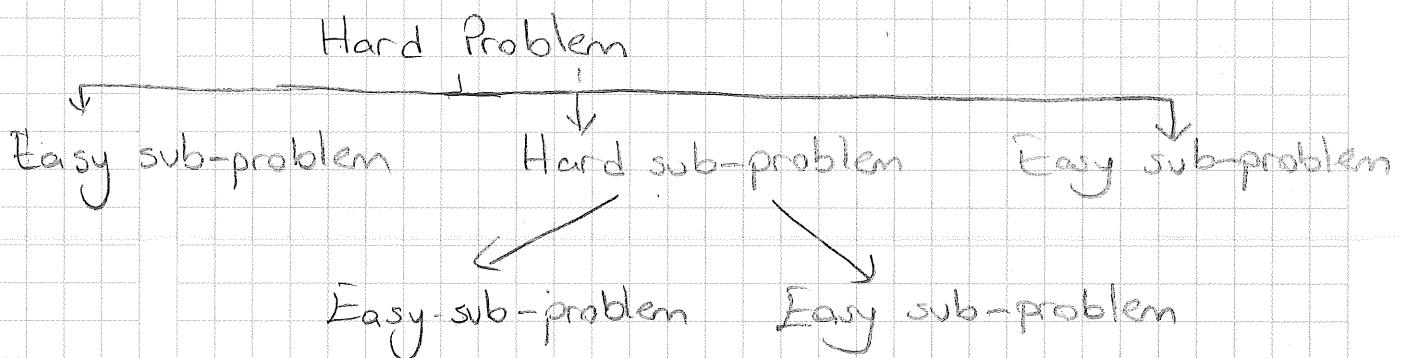
④ Algorithms: Designing and implementing algorithms.

What is programming?

Programming = Algorithms + Coding

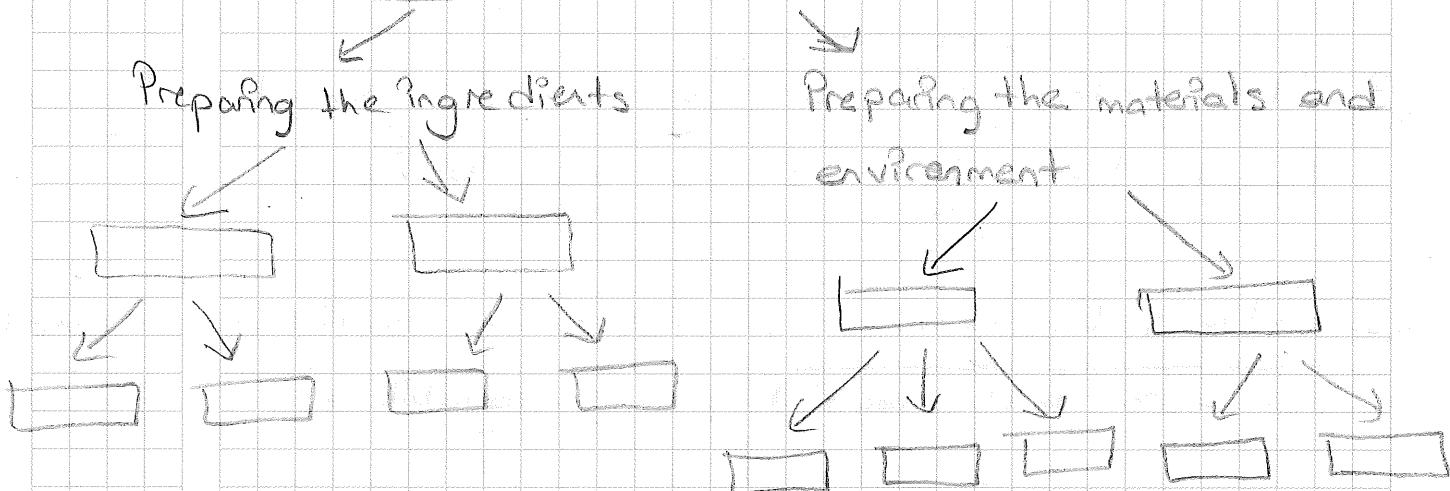
Ayrışma

① DECOMPOSITION



We use tree structures to visualize decomposition
(Belli başlıklar altında kategorize ederek görselleştirmek)

Making a Chocolate Cake



An iteration is the repe

During art class, I created 2 _____
of a drawing.

(*) Computer programming is a set of instructions to get the result you want.

(*) Loops repeat a set of instructions.

↳ Tekrar ettirme (Belki 100 defa aynı şeyi tekrar ettirme)

(*) Sequence: Koniut sırası.

② PATTERN RECOGNITION:

Pattern recognition is recognising patterns in different things (solutions) and being able to apply these patterns in the solutions of other similar problems.

Pattern recognition is good because we don't have to come up with a new solution each time we encounter the same problem. Instead, we apply the patterns that we have previously recognised.

(4) ALGORITHM DESIGN: ^{Kodlamadan önceki son aşamayı} Açık, ayrıntılı, aştalı entegratif gibi olmalı.

* An algorithm is a sequence of clearly defined steps to describe a process.

* Algorithms are useful when we wish to explain someone else or a computer how to carry out steps.

* Algorithms are important for programming because a correct algorithm is the ultimate basis of any computer-based solution.

COFFEE ALGORITHM

Filtre kahveyi al.

1. ölü filter kahve al (ölü kabı al)

Filtre kahve makinesinin kahve bölümüne, 1. ölü kahveyi ekle.

Kahve eklenen bölümün kapat, ölü kabını bırak.

Su ölü kabını al, musluktan 1 bardak su doldur.

Filtre kahve makinesinin su hanesine suyu ilave et.

Kahve bardağını servis bölümünde koy.

Kahve makinesinin fırını piye tak.

③ ABSTRACTION → İhtiyaçın olmayan detayları kaldır.

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.

is tanımı

PSEUDOCODE

Computational thinking'in kod yazılmadan önceki son aşaması.

Teknik resim bir defa çizilir ve bütün inşaat mühendisleri bunu anlar. Herkes farklı teknik kullanabilir (Örneğin programcılarrawn bir Python ile Java kullanabilir.)

Amaç standartlığı sağlanmaktadır.

Algortmanın ifade edilmesidir. Programlama diline yakındır ama programlama dilini bilmeyen programcı anlar.

Kullanıcıdan sunu isterin:

What's your name?

Kullanıcıdan sunu isterin:

How old are you?

if age >= 70 then

'You are aged to perfection!'

else

'You are a spring chicken!'

Bunu Passenger Python'a

yazar ister Java'ci

Bu bir pseudocode'dur.

KEYWORDS → Bunları cümle içinde kullanma

, START, BEGIN

INPUT

READ, GET

PRINT, DISPLAY, SHOW, OUTPUT

TÜRK KAHVESİ İÇİN PSEUCODE

BEGIN

OUTPUT

INPUT1 : Kas Kışlık

INPUT2 : Şekerli - Sadık

SET : Türk Kahvesi

Su
Şeker
Cezve
Kasık
Fincan
Bardak

SET INPUT1 : Fincan

Bardak

SET : Ocak

Cezve

if INPUT2 : Şekerli

 add Kahve + Sut + Şeker

else add sut + Kahve

OUTPUT : Türk Kahvesi

END

BEGIN

INPUT hours, rate

IF hours < 40

THEN

 pay = hours * rate

ELSE

 pay = 40 * rate + (hours - 40) * rate * 1.5

OUTPUT pay

END

Degiskenlerin başına SET koy.

Birdeñ farla şart olursa

IF - ELSEIF - ELSE → Discards kalan tüm durumlar

Her
prog.
kendisi
birdeñ
göre
yapar.

{ 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.

FOR STRUCTURE

For loop runs for each element inside a group.

For example:

For every day of the week

Count;

endfor

For every 25 minutes of study

Earn one Pomodoro; → Pomodoro = Pomodoro!

endfor

tongue

```
list = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
for x in list:
```

```
    print(x)
```

```
    value = (x ** 2 + 5)
```

```
    print(value)
```

```
    print("exit the for loop")
```

```
for i in range(25, 100, 5):
```

25'den 100'e 5'ler adımlarla
git.

REPRESENT AN ALGORITHM

Pseudocode

Flowchart

Pseudocode. Is an informal high-level description of a computer program or algorithm. It's written in symbolic code which must be translated into a programming language before it can be executed. Using Pseudocode is similar to "writing in a programming language" and might look something like this.

OUTPUT 'What is your name?'

INPUT 'User inputs his/her name.'

STORE the user's input in the name variable.

OUTPUT 'Hello' + name

OUTPUT 'How old are you?'

INPUT user inputs his/her 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!'

INPUT → Ask a question

OUTPUT → Prints a message on the screen.

► Pseudocode is understood by the programmers of all types

► It enables programmer to concentrate only on the algorithm part of the code development.

► It cannot be compiled into an executable program.

MATHEMATICAL OPERATIONS

Assignment : ← or :=

Example → $c \leftarrow 3, c := 2$

Comparison : =, ≠, <, >, ≤, ≥

Arithmetic : +, -, ×, /, mod

Logical : and, or

KEYWORDS

START, BEGIN → Start of your pseudocode

INPUT → Data retrieved from the user

READ, GET → Used when reading data from a data file.

PRINT, DISPLAY, SHOW, OUTPUT → 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 → End of the pseudocode

IF - IF ELSE - ELSE

That is used to provide statements to be executed if a certain condition is met.

CASE

It is used if we want to compare a single variable against several conditions.

INPUT color

CASE color of

red: PRINT "red"

green: PRINT "green"

blue: PRINT "blue"

OTHERS

PRINT "Please enter a value
color"

END CASE

The OTHERS clause with its statement is optional. Conditions are normally numbers or characters.

Iteration: Repeat a set of instructions

→ Yinelemek

FOR

Listenin eleman sayisina bagli olarak yapilir.

Takes a group of elements and,

Runs the code within the loop for each element.

FOR every month in a year

Compute number of days

ENDFOR

WHILE → Boolean bit degit var. Verilen digerinden boykicik olma durumuna gore true-false

Repeat a block of code (like for)

How long the condition will remain true (unlike for)
(Kosul ne kadar süre doğru kalacak?)

PRECONDITION: variable X is equal to 1

WHILE Population < Limit

Compute Population as Population + Births - Deaths.

ENDWHILE

Repeat until → Yondaki verilen oldigunda "program" sonlanadir. (while' dan farki)

repeat until = while not

scratch

Python

Algorithm: Is the steps you must take to achieve a specific goal.

FUNCTIONS

When solving advanced tasks it is necessary to break down the concepts in a block of statements in different locations. To reuse this code, we create functions. We can then call these functions every-time we need them to run.

Function clear monitor

Pass In : nothing

Direct the operating system to clear the monitor.

Pass Out : nothing

Endfunction

To emulate a function call in pseudocode, we can use the Call keyword

call : clear monitor

* Algoritma Aksi : Sequence

Selection

Iteration

PROGRAM WRAPPING

We wrap our code as a program. A program can be defined as a set of instructions that performs a specific tasks when executed.

```
PROGRAM makeacupoftea  
END
```

EXCEPTION HANDLING (İstisnai İşleme)

Akış, bozun, istenmeyen olaylar. Bu olayları yönetmek ve bunlara yanıt olarak kod blokları oluşturmak gerekiyor.

```
BEGIN
```

```
statements
```

```
EXCEPTION
```

```
WHEN exception type
```

```
statements to handle exception
```

```
WHEN another exception type
```

```
statements to handle exception
```

```
END
```

CONCLUSION (Sonuç)

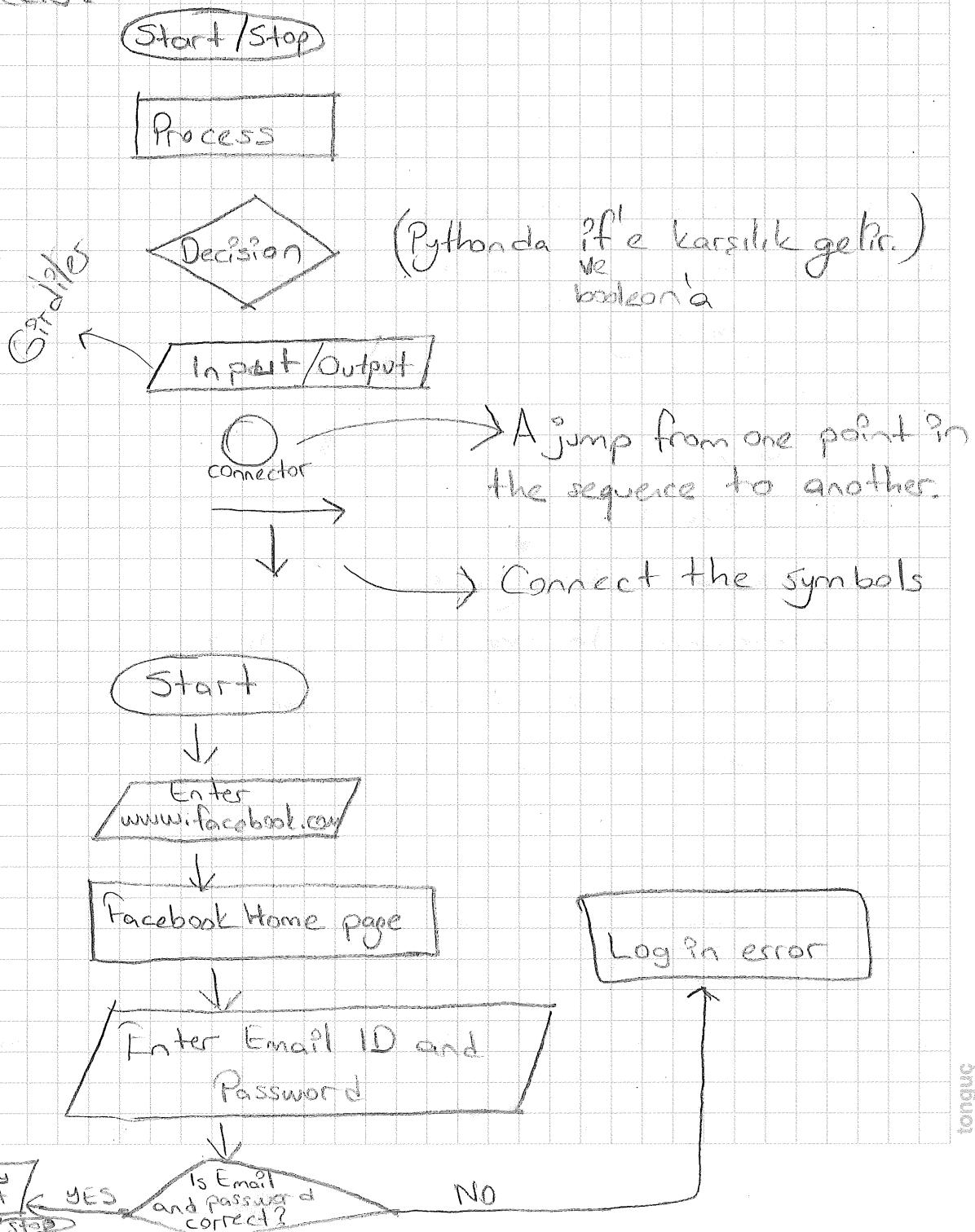
Pseudocode'ın kesin kuralları yok, intanlar okuya bilmem gerek. Ama yine de bazı kılavuzlar vardır.

- Fortran style pseudocode
- Pascal style
- C style

FLOWCHART

Okularla ifade edilen algoritma.

Flowchart in her bir elemanı konut singingini
tersil eder.



tongue

4.

1.

2.

