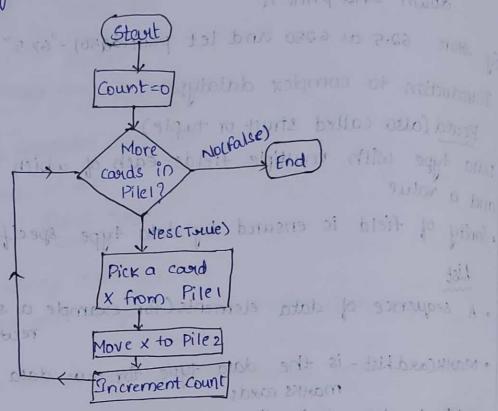
Week-2

Flowchoots

Pictorial supresentation of computational prod

remitted of from the substitutions

Eg: counting the number of cards



1) Process (Indicated in Rectargular box) -> Node types.

@ Decision (Indicated in Diamond shape box)

3) Terminal ( used for stant and end. Indicated by C -> Arronis indicate, operation flow

Poros and coms of "flowichants" and sail so "Harokist is a list of Itano data type

Advantages:

- · Visual suppresentation of computation surer and asnitab east ofthe
- · Easy to understand

- Disadvantages:

  Size: Complex processes generate Large flowthauts
  - · Collaboration: Shooting pictures in editable format
    - · versions: Companie changes between flouichants

```
Follom pictures to text.
 Decribe the previous process in words
Step 0 Start
Step 1 Initialize Count to o
Step 2. Check courds in Pile 1
Step 3 If no more couds, go step 8
Step 4 Pick a cand X from Pile 1
Step 5 Move X to Pile 2 statement and parameters
Step 6 Increment Count and brighting.
Step 7 Go back to step 2
Purguamming language boro moitored la standouse
Step 8 End
-> Succinct notation for computational processes
-> Better textual supprecentation for Conditional execution
 Step 3 If no more conds, go to step 8
 ostep 4 Pick a could x from Pite 1
 Repeated execution
 Step 2 Check cands in Pile 1
 Step 7 Gio back to Step 2
Pseudocode
Start
                        till a rand of foren filt !
Count = 0
 while (Pile a 1 has more conds) {
                          Sinn - Sum + Sharps
 Pick a could X from Pile 1
 Move x to Pile 2
 Increment Count
```

- O Assign a value to a vooliable @ Repeat steps while condition holds 3 Mark start and end of suppeated block
- Summary
- Flourchasts are easy to read, visual descriptions of procedures
- ... but they are cumbersome, hard to share and edit.
- Writing down steps in a text is an alternative
- Tune the notation to capture standard features
  - · Assigning values to variables freed formstores &
  - · Conditional execution
  - · Repeated execution

## Pseudocode: Iteration and filtering

· Counting courds and tometologones not mother travitibant us collections per lecture

Start

Count = D

while CPile 1 has more coulds) {

Pick a coud x from Pile 1

Move X to Pile 2

Increment Count

End

- Will dispense with Start and End henceforth

Sum of Maths marks

Sumon While (Pile 1 has more coulds) { Pick a card X from Pile 1 Move X to Pile 2 Sum = Sum + XMaths

(Stout) sum=0 End coulds in J. Yes (Town) Pick a could x from Pile! Move x to Pilez Sum = Sum + x · Maths

- Update Sum: assignment statement
   Sum on suight is current value
   Sum on left is updated value
   = is not mathematical equality
- Increment: Count = Count +1
- \* x Maths : Maths marks in cord x

Sum of Boys' Maths marks

Sum = 0

while (Pile 1 has more cands) {

Pick a cond x from Pile 1

Move X to Pile 2

if (xGrender = = m) {

Sum = Sum + x Maths

}

- a Conditional execution once.
- Equality (==) vs assignment (=)

Sum of Boys' and Girls' Maths marks

Boy Sum = 0

Girl Sum = 0

While (Pile 1 has more cands) {

Pick a cand X

Move X to Pile 2

if ( Gender = = m) {

Boy Sum = Boy sum + x Maths

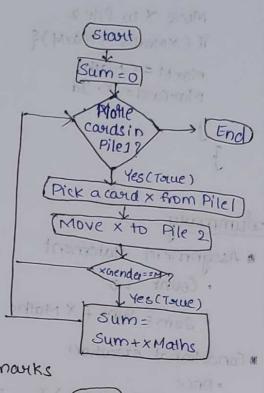
}

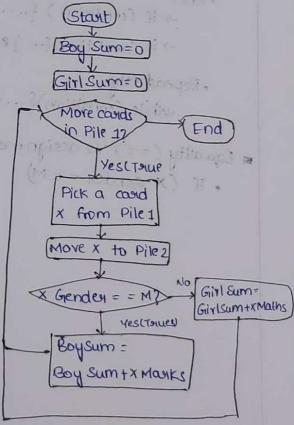
else {

Girlsum = Girlsum + X Maths

}

alternative branch for conditional





Finding the card with maximum Maths mounts

Max M = 0

Max card = -1

While (Pile 1 has more cards) {

More card x from Pile 1

More x to Pile 2

If (x Mounts > Max M) {

Max M = x Maths

Max Card = X. Id

Move x to F

## Summory

- · Assignment statement
  - · Count = 0
  - · Sum = Sum + X Maths.
- · Conditional execution
  - · Once

-> if (condition) f .... }

→ if (condition) {....} else {....}

· Repeatedly while (condition) § ... 3

■ Equality (==) vs assign ment(=) • if (x Gender == M) Max M = N Mak M

Yes Chrue

Pick a cond x

from Pile 1

Move x to Pile 2

No

X Maths > Max M

Yes Chrue

Max M = X Maths

Max M = X Maths

Max M = X Maths

of Ends, and Public Wallie

in william are record with many

Will (File 1 has man cases)

X bress cars