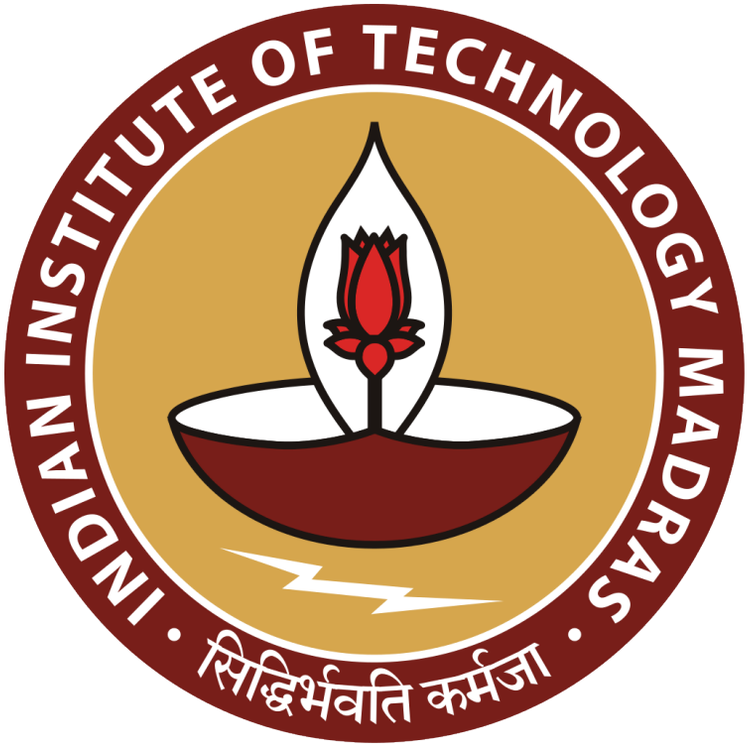


IIT Madras

ONLINE DEGREE

Pseudocode: Iteration and Filtering

Counting cards

Start

Count = 0

while (Pile 1 has more cards) {

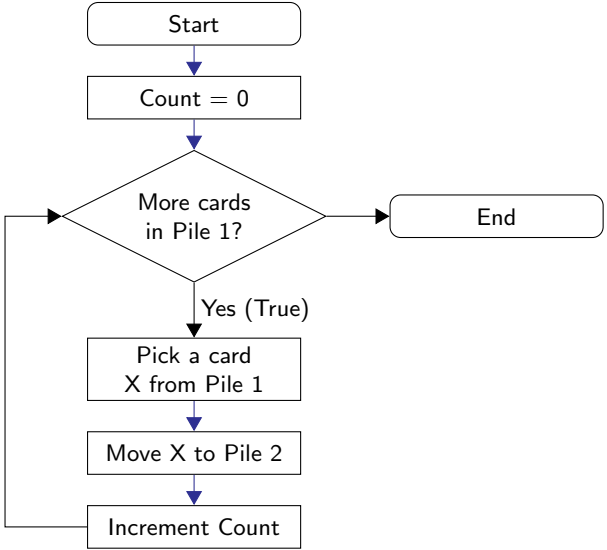
  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Increment **Count**

}

End



Counting cards

Start

Count $= 0$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

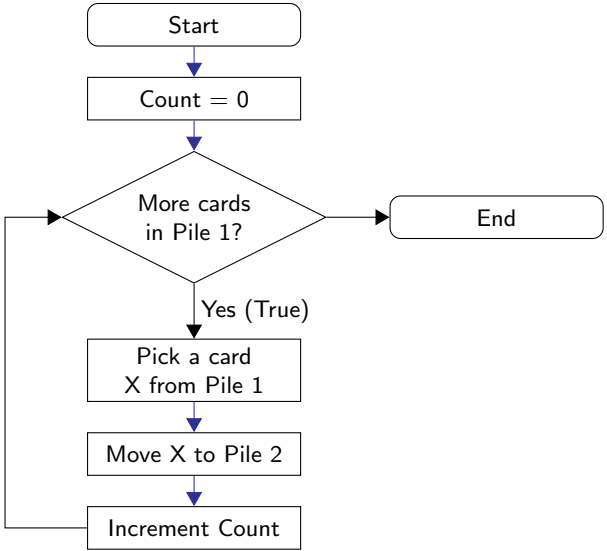
  Move **X** to Pile 2

  Increment **Count**

}

End

- Will dispense with Start and End, henceforth



Sum of Maths marks

Count = 0

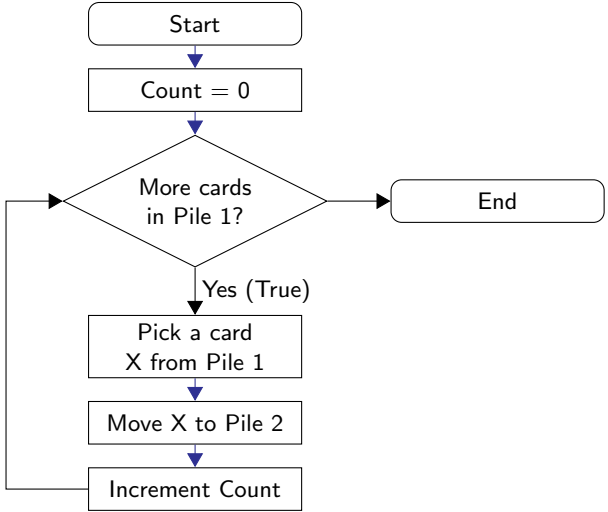
while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Increment **Count**

}



Sum of Maths marks

Sum = 0

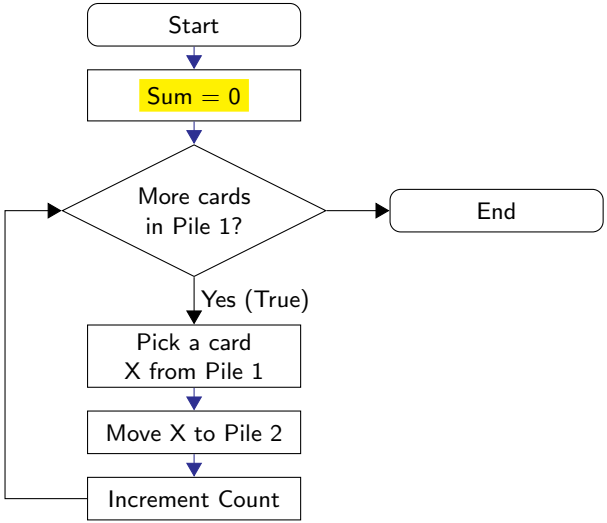
while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Increment **Count**

}



Sum of Maths marks

Sum = 0

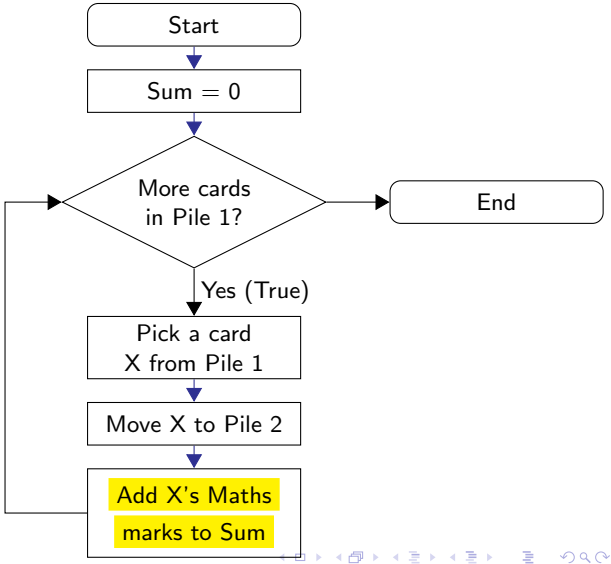
while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Add **X**'s Maths marks to **Sum**

}



Sum of Maths marks

Sum $= 0$

while (Pile 1 has more cards) {

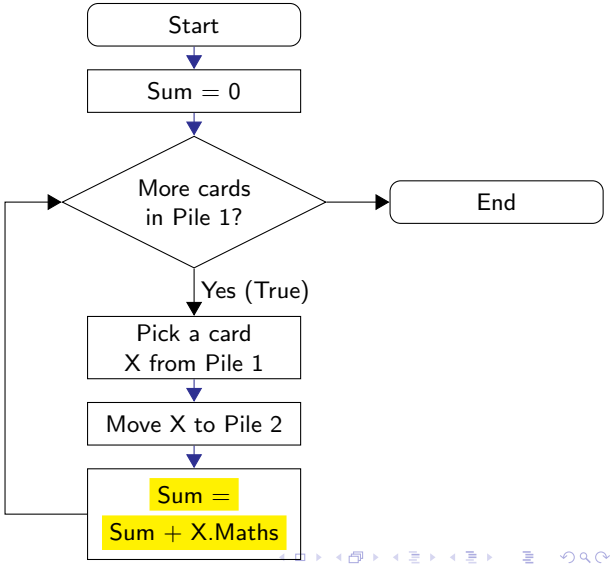
  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Sum $=$ **Sum** $+$ **X**.Maths

}

- Update **Sum** : assignment statement
  - **Sum** on right is current value
  - **Sum** on left is updated value
  - $=$ is not mathematical equality!



Sum of Maths marks

Sum $= 0$

while (Pile 1 has more cards) {

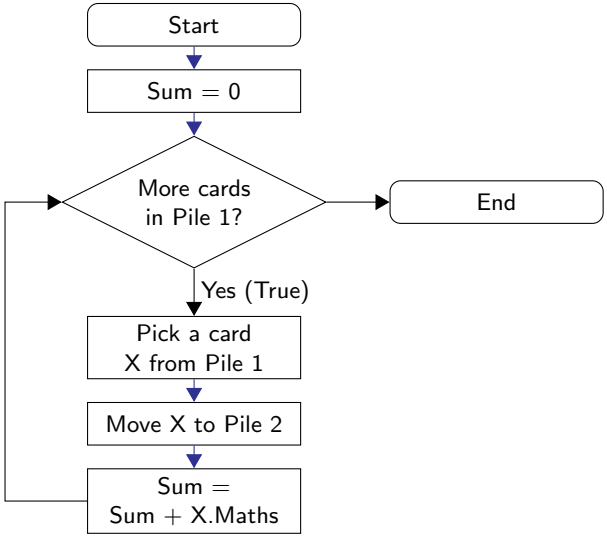
  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Sum $=$ **Sum** $+$ **X**.Maths

}

- Update **Sum** : assignment statement
  - **Sum** on right is current value
  - **Sum** on left is updated value
  - $=$ is not mathematical equality!
- Increment: **Count** $=$ **Count** $+ 1$



Sum of Maths marks

Sum $= 0$

while (Pile 1 has more cards) {

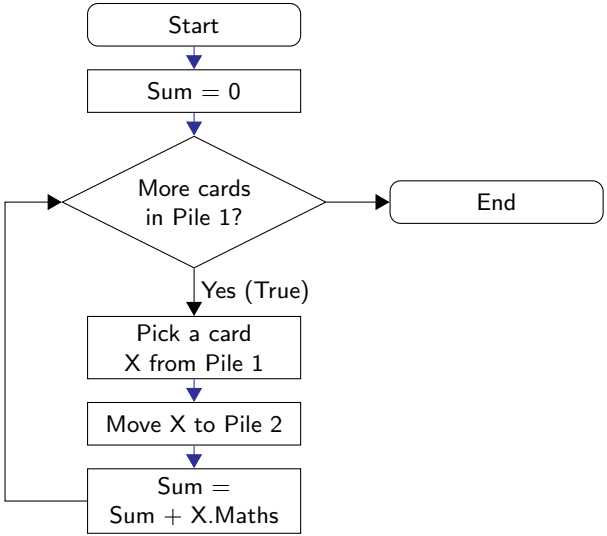
  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Sum $=$ **Sum** $+$ **X**.Maths

}

- Update **Sum** : assignment statement
  - **Sum** on right is current value
  - **Sum** on left is updated value
  - $=$ is not mathematical equality!
- Increment: **Count** $=$ **Count** $+ 1$
- **X**.Maths : Maths marks in card **X**



Sum of Maths marks

Sum $= 0$

while (Pile 1 has more cards) {

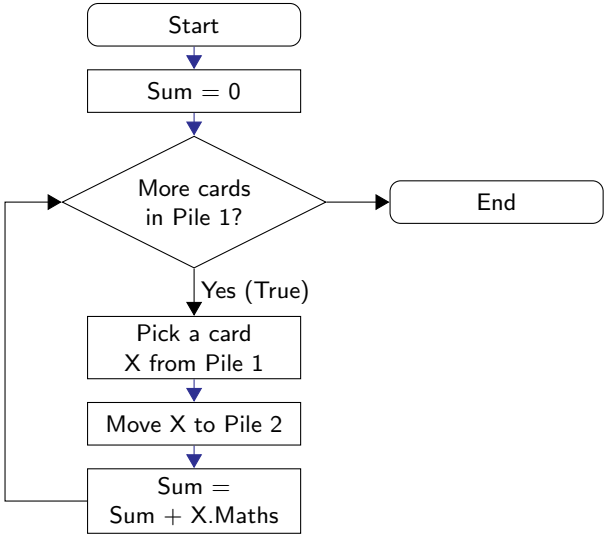
  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Sum $=$ **Sum** $+$ **X**.Maths

}

- Update **Sum** : assignment statement
  - **Sum** on right is current value
  - **Sum** on left is updated value
  - $=$ is not mathematical equality!
- Increment: **Count** $=$ **Count** $+ 1$
- **X**.Maths : Maths marks in card **X**



Sum of Boys' Maths marks

Sum = 0

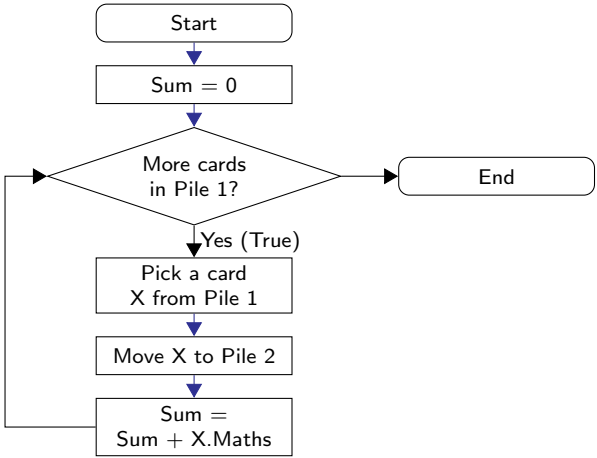
while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Sum = **Sum** + **X**.Maths

}



Sum of Boys' Maths marks

Sum = 0

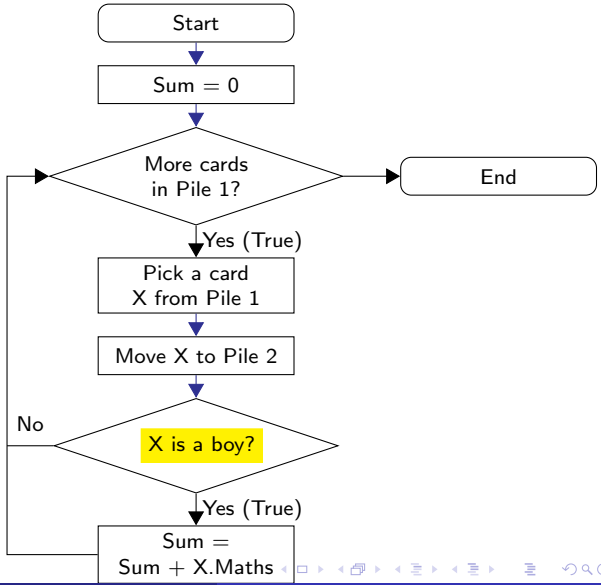
while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

  Sum = **Sum** + **X**.Maths

}



Sum of Boys' Maths marks

Sum = 0

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

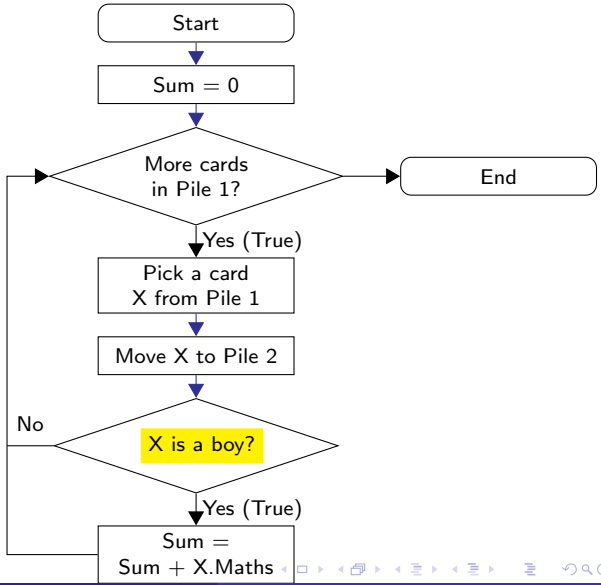
  Move **X** to Pile 2

  if (**X** is a boy) {

    Sum = **Sum** + **X**.Maths

  }

}



Sum of Boys' Maths marks

Sum $= 0$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

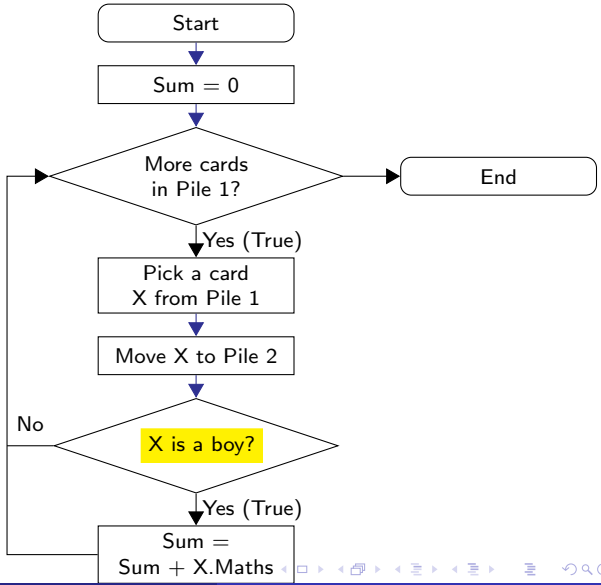
  if (**X** is a boy) {

    Sum $=$ **Sum** $+$ **X**.Maths

  }

}

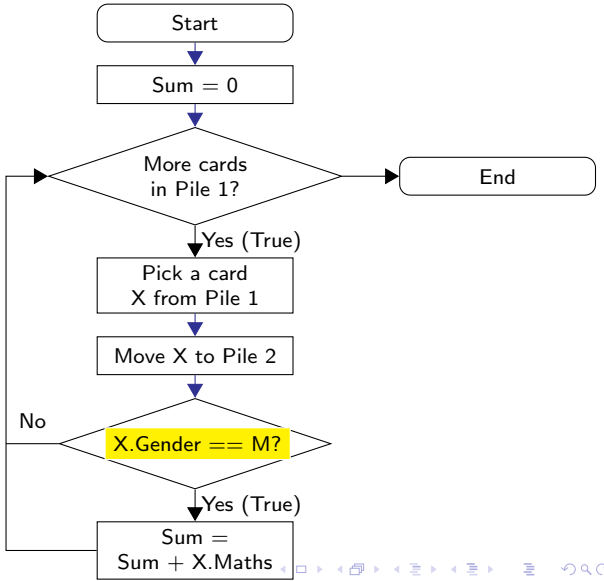
- Conditional execution, once



Sum of Boys' Maths marks

Sum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    Sum = **Sum** + **X**.Maths
  }
}

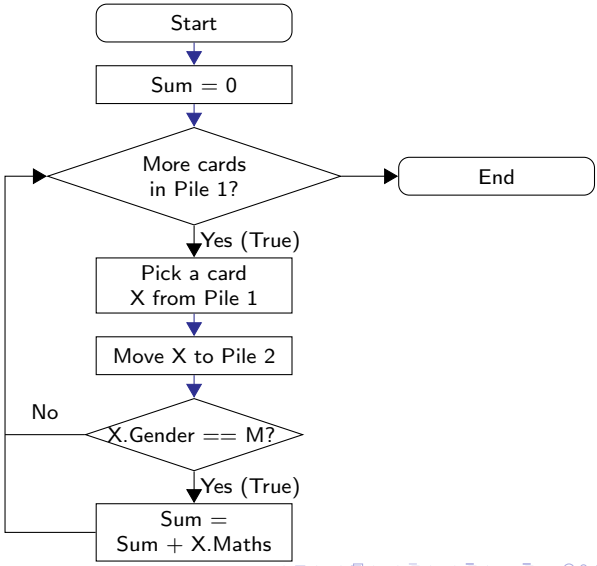
- Conditional execution, once
- Equality (==) vs assignment (=)



Sum of Boys' Maths marks

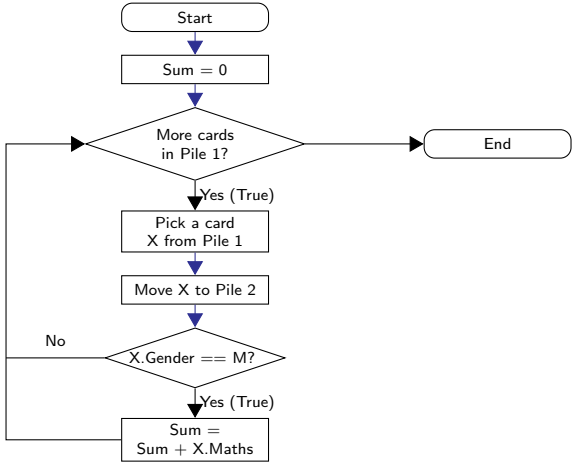
Sum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    Sum = **Sum** + **X**.Maths
  }
}

- Conditional execution, once
- Equality (==) vs assignment (=)



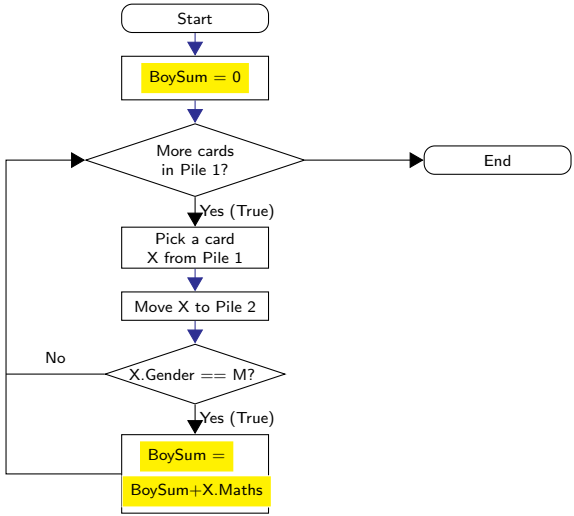
Sum of Boys' and Girls' Maths marks

Sum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    Sum = **Sum** + **X**.Maths
  }
}



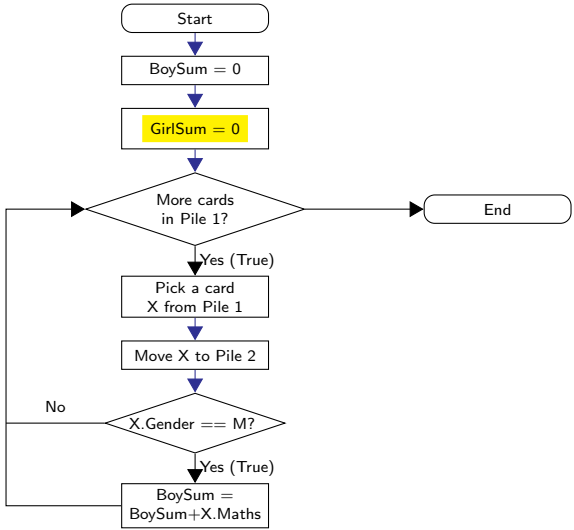
Sum of Boys' and Girls' Maths marks

BoySum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    BoySum = **BoySum**+**X**.Maths
  }
}



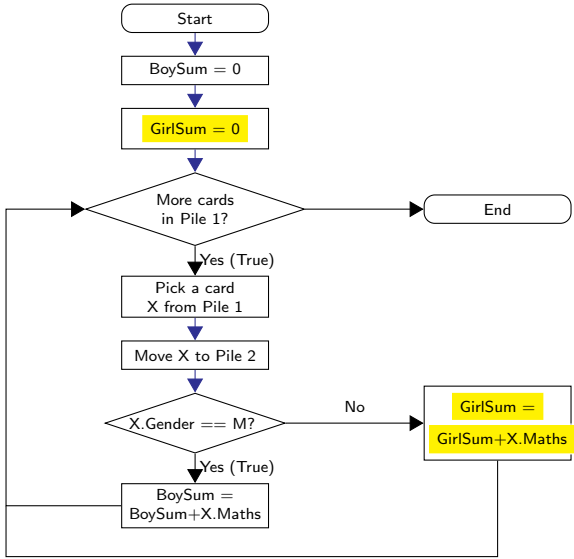
Sum of Boys' and Girls' Maths marks

BoySum = 0
GirlSum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    BoySum = **BoySum** + **X**.Maths
  }
}



Sum of Boys' and Girls' Maths marks

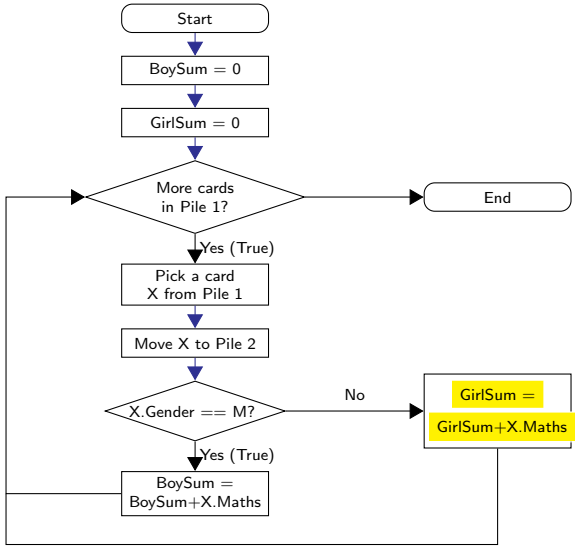
BoySum = 0
GirlSum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    BoySum = **BoySum** + **X**.Maths
  }
  else {
    GirlSum = **GirlSum** + **X**.Maths
  }
}



Sum of Boys' and Girls' Maths marks

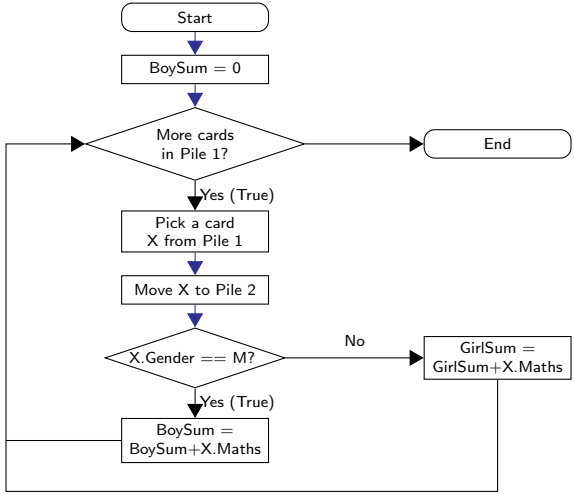
BoySum = 0
GirlSum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    BoySum = **BoySum** + **X**.Maths
  }
  else {
    GirlSum = **GirlSum** + **X**.Maths
  }
}

- Alternative branch for conditional



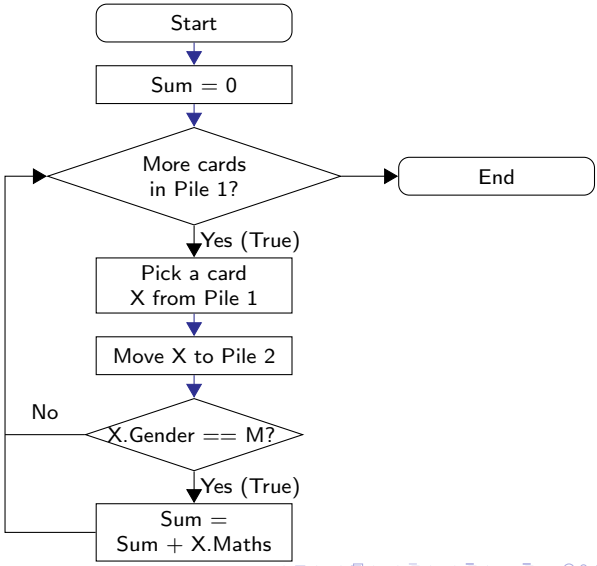
Sum of Boys' and Girls' Maths marks

BoySum = 0
GirlSum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    BoySum = **BoySum** + **X**.Maths
  }
  else {
    GirlSum = **GirlSum** + **X**.Maths
  }
}



Finding the maximum Maths marks

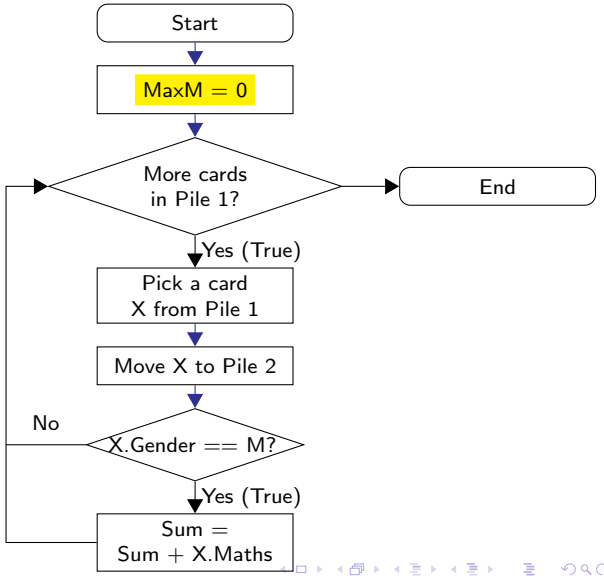
Sum = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Gender == M) {
    Sum = **Sum** + **X**.Maths
  }
}



Finding the maximum Maths marks

MaxM = 0

```
while (Pile 1 has more cards) {
  Pick a card X from Pile 1
  Move X to Pile 2
  if (X.Gender == M) {
    Sum = Sum + X.Maths
  }
}
```



Finding the maximum Maths marks

MaxM $= 0$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

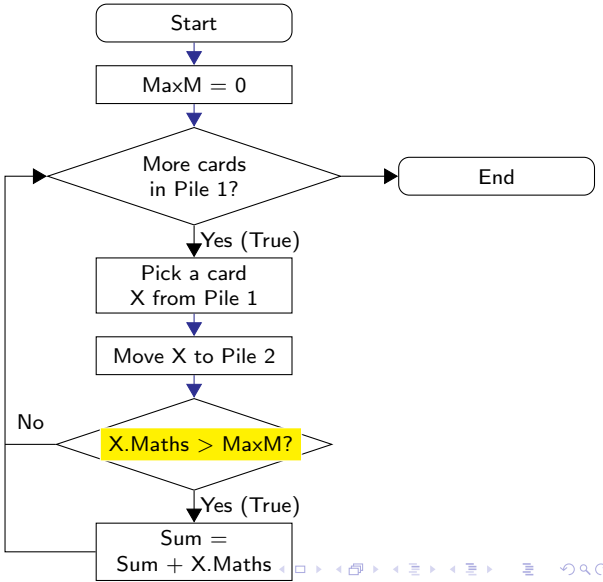
  Move **X** to Pile 2

  if (**X**.Maths $>$ **MaxM**) {

    Sum $=$ **Sum** $+$ **X**.Maths

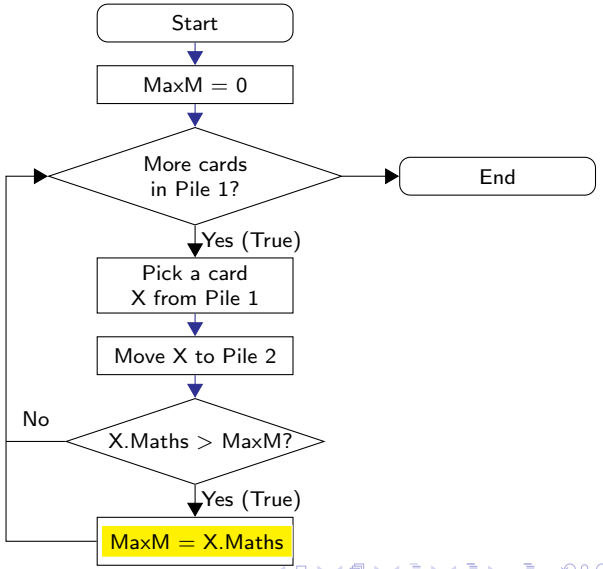
  }

}



Finding the maximum Maths marks

MaxM = 0
while (Pile 1 has more cards) {
  Pick a card **X** from Pile 1
  Move **X** to Pile 2
  if (**X**.Maths > **MaxM**) {
    MaxM = **X**.Maths
  }
}



Finding the maximum Maths marks

MaxM $= 0$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

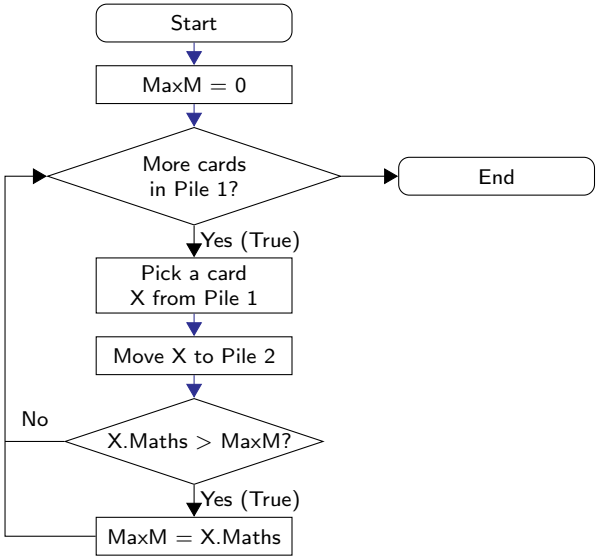
  Move **X** to Pile 2

  if (**X**.Maths $>$ **MaxM**) {

    MaxM $=$ **X**.Maths

  }

}



Finding the card with maximum Maths marks

MaxM $= 0$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

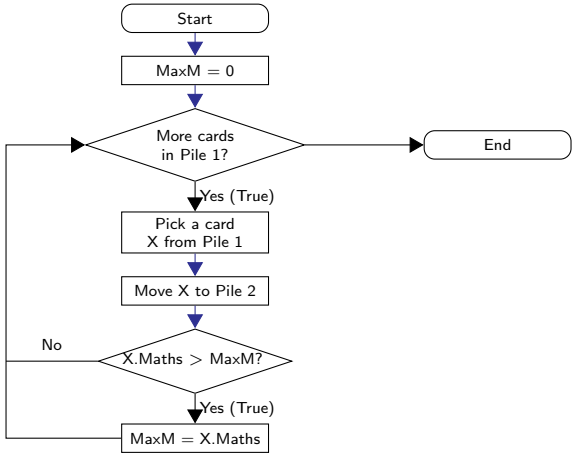
  Move **X** to Pile 2

  if (**X**.Maths $>$ **MaxM**) {

    MaxM $=$ **X**.Maths

  }

}



Finding the card with maximum Maths marks

MaxM $= 0$

MaxCard $= -1$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

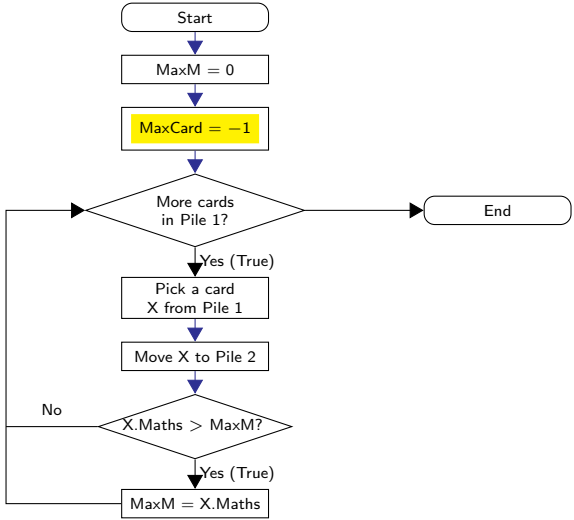
  Move **X** to Pile 2

  if (**X**.Maths $>$ **MaxM**) {

    MaxM $=$ **X**.Maths

  }

}



Finding the card with maximum Maths marks

MaxM $= 0$

MaxCard $= -1$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

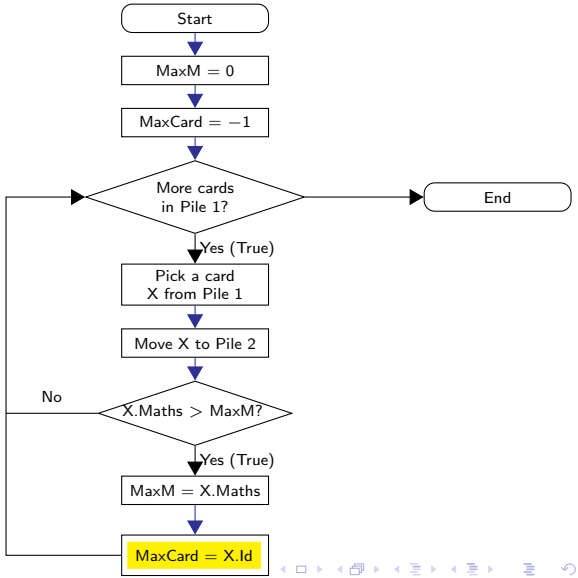
  if (**X**.Maths $>$ **MaxM**) {

    MaxM $=$ **X**.Maths

    MaxCard $=$ **X**.Id

  }

}



Finding the card with maximum Maths marks

MaxM $= 0$

MaxCard $= -1$

while (Pile 1 has more cards) {

  Pick a card **X** from Pile 1

  Move **X** to Pile 2

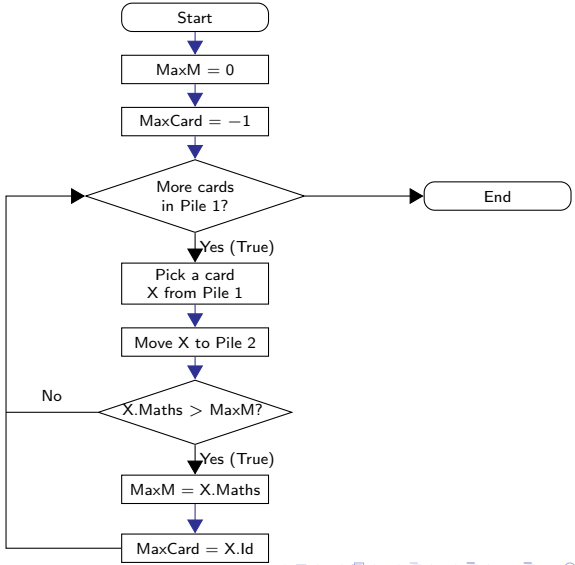
  if (**X**.Maths $>$ **MaxM**) {

    MaxM $=$ **X**.Maths

    MaxCard $=$ **X**.Id

  }

}



Summary

- Assignment statement

Summary

- Assignment statement
  - **Count** $= 0$

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths
- Conditional execution

Summary

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

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths
- Conditional execution
  - Once
    - if (condition) { . . . }

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths
- Conditional execution
  - Once
    - if (condition) { . . . }
    - if (condition) { . . . } else { . . . }

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths
- Conditional execution
  - Once
    - if (condition) { . . . }
    - if (condition) { . . . } else { . . . }
  - Repeatedly

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths
- Conditional execution
  - Once
    - if (condition) { . . . }
    - if (condition) { . . . } else { . . . }
  - Repeatedly
    - while (condition) { . . . }

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths
- Conditional execution
  - Once
    - if (condition) { . . . }
    - if (condition) { . . . } else { . . . }
  - Repeatedly
    - while (condition) { . . . }
- Equality ($==$) vs assignment ($=$)

Summary

- Assignment statement
  - **Count** $= 0$
  - **Sum** $=$ **Sum** $+$ **X**.Maths
- Conditional execution
  - Once
    - if (condition) { . . . }
    - if (condition) { . . . } else { . . . }
  - Repeatedly
    - while (condition) { . . . }
- Equality ($==$) vs assignment ($=$)
  - if (**X**.Gender $==$ M)