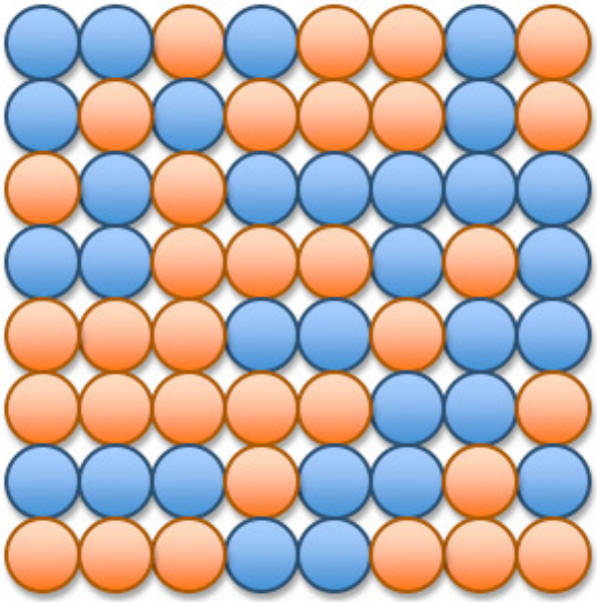


UNIT 3C
Algorithmic thinking continues

Announcements

- PS2 is due Today Friday Feb 1 in class.
- PA3 (2/5) and PS3 (2/8) are out now
  - Topics covered
    - conditionals
    - Iterations and use of \ and %
    - ASCII art – nested loops
    - Counting and while loops
    - Tracing code
    - From algorithm to code
    - Flow charts

Return Statement

- A return statement is just like a print statement TRUE/FALSE
- What is the purpose of having a return statement in a function? to save the value for future use
- Can we have more than one return statement in a function?

yes but only one can return

```
if (x > y) then
    return x
else
    return y
end
```

Return Statement ctd..

- Can we return more than once from a function?

No

```
...
return x
...
return y
```

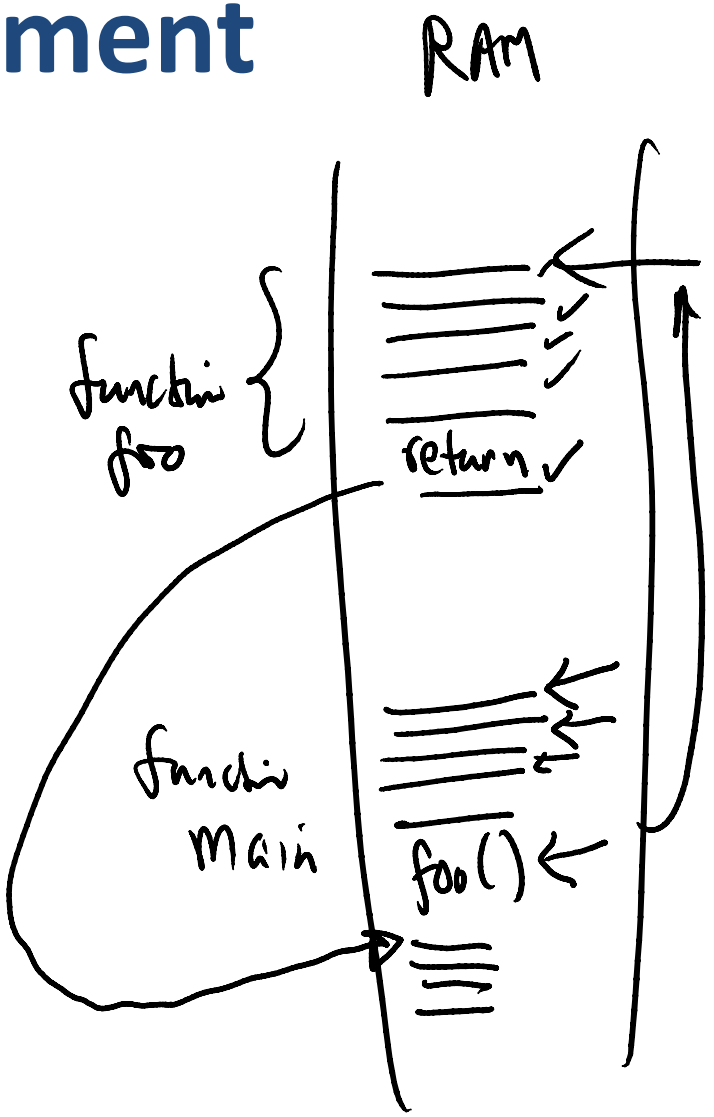
- Can we have a return statement inside a for or while loop?

NO

```
for i in 1..10 do
    print i
    return i
end
```

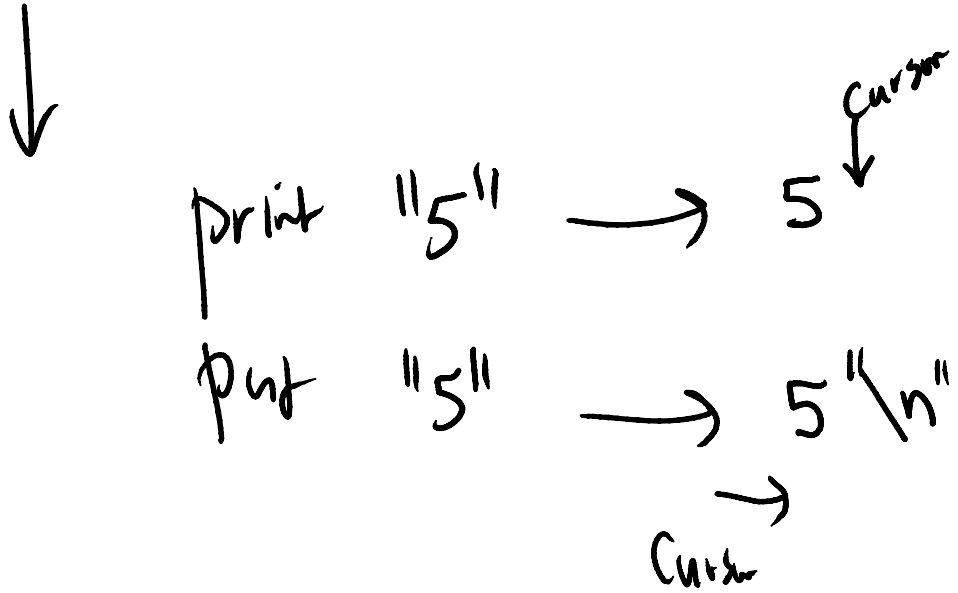
Return statement

- How does it work?



Questions??

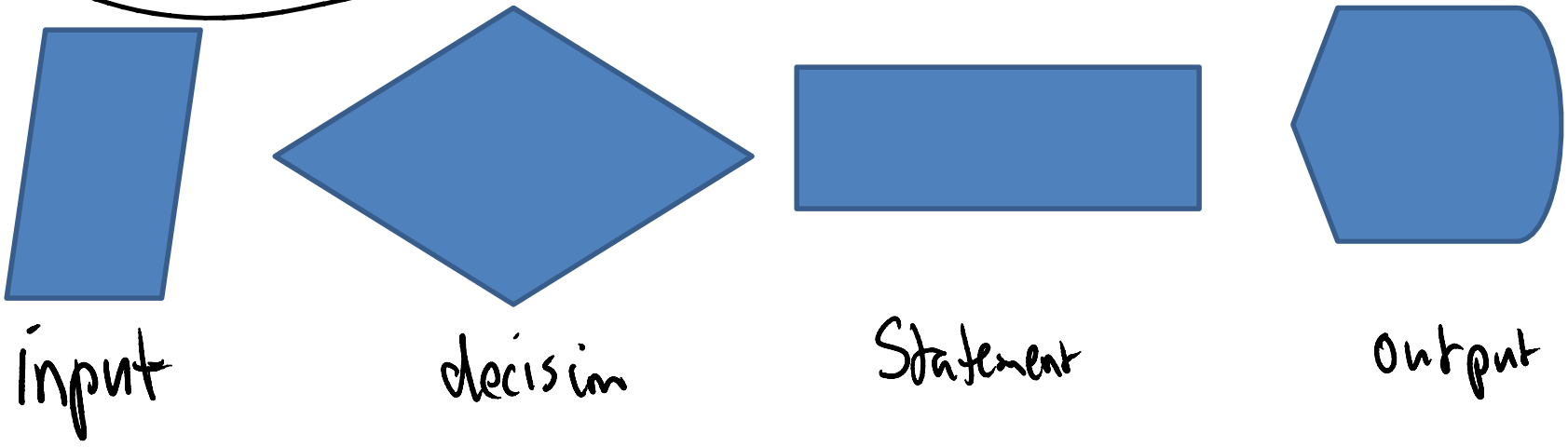
- What is the difference between PRINT and PUT statements in Ruby?



Tools for Developing algorithms

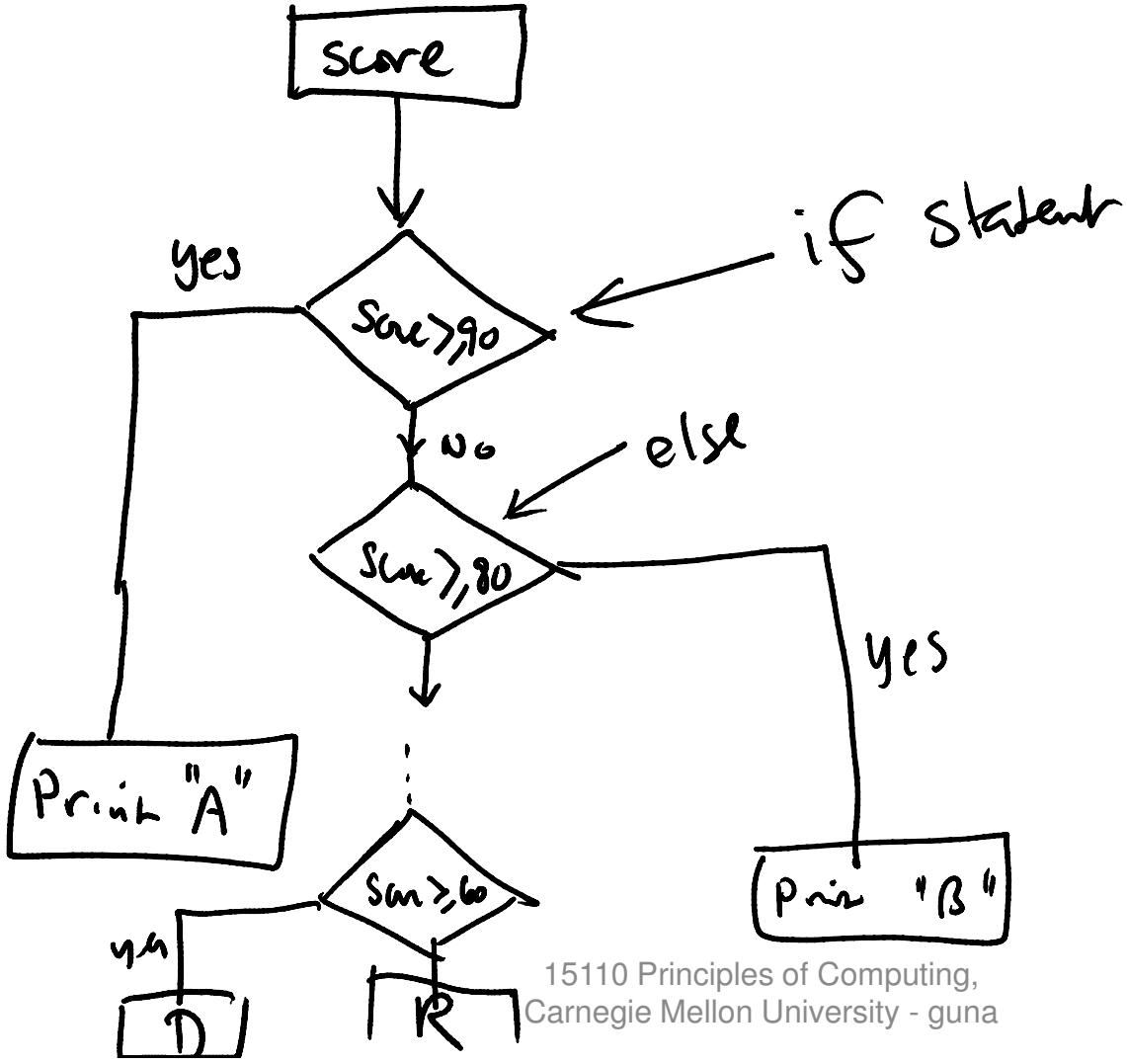
Developing algorithms

- An algorithms must be
  - correct, efficient, tested
- Tools for developing algorithms
  - Flow Charts



Draw a flow chart

- **Given a score, print a grade**



Tools for Implementing algorithms

Truth Tables

AND, OR, NOT tables

AND	T	F
T	T	F
F	F	F

OR	T	F
T	T	T
F	T	F

NOT	T	F
	F	T

- DeMorgans Law

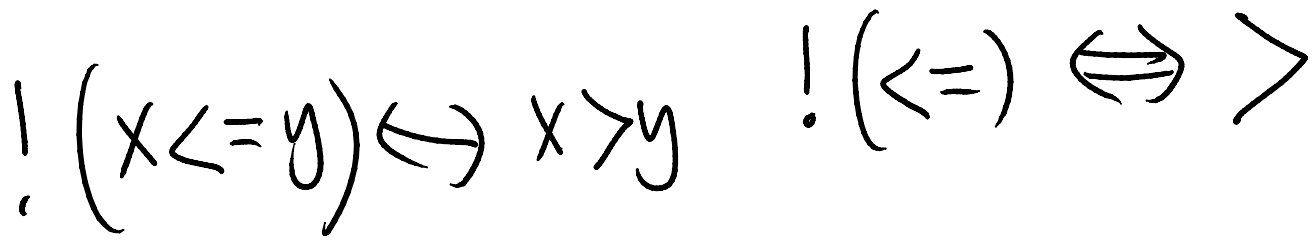
!(A AND B) = !A or !B

!(A or B) = !A and !B

Relational Operators

- If we want to compare two integers to determine their relationship, we can use these relational operators:

<	less than	<=	less than or equal to
>	greater than	>=	greater than or equal to
==	equal to	!=	not equal to



Examples

- Classify following statements as “always false” “always true” or if “sometimes true”, then provide a value(s).

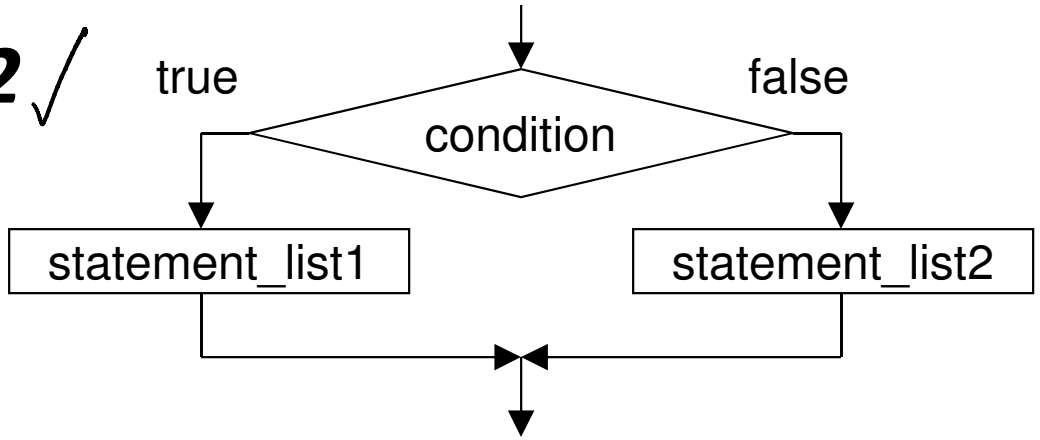
statement	Always false	Always true	Sometimes true
1 == 0	✓		
x >= 1 or x < 1		✓	
x >= 1 and x < 1	✓		
x >= 1 or x < -2			~~x <= 0~~ 30 ✓

Branching

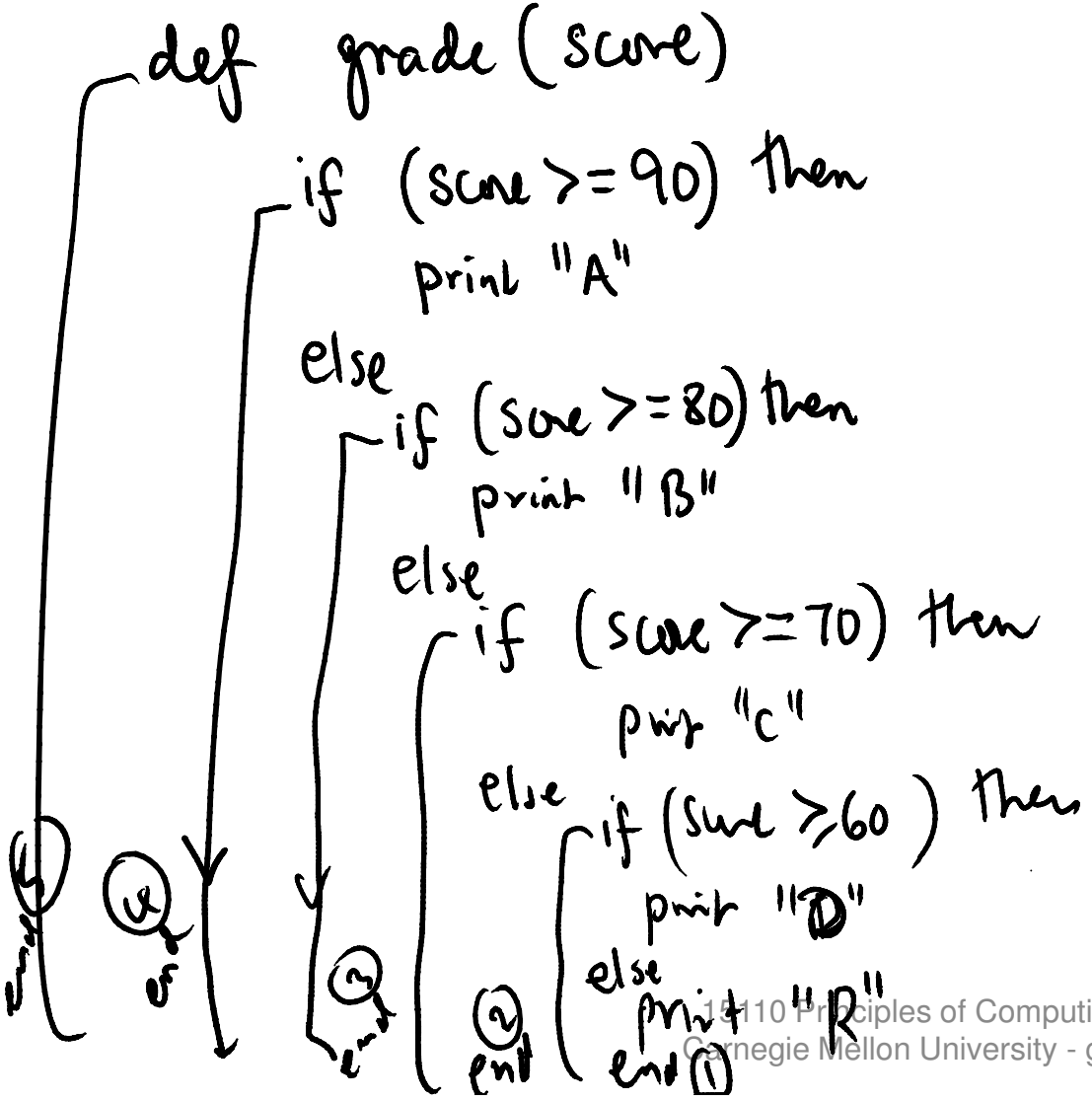
`if/else` statement

Format:

```
if bool_condition then
    statement_list1
else
    statement_list2
end
```



Write a function to print the grade given a score



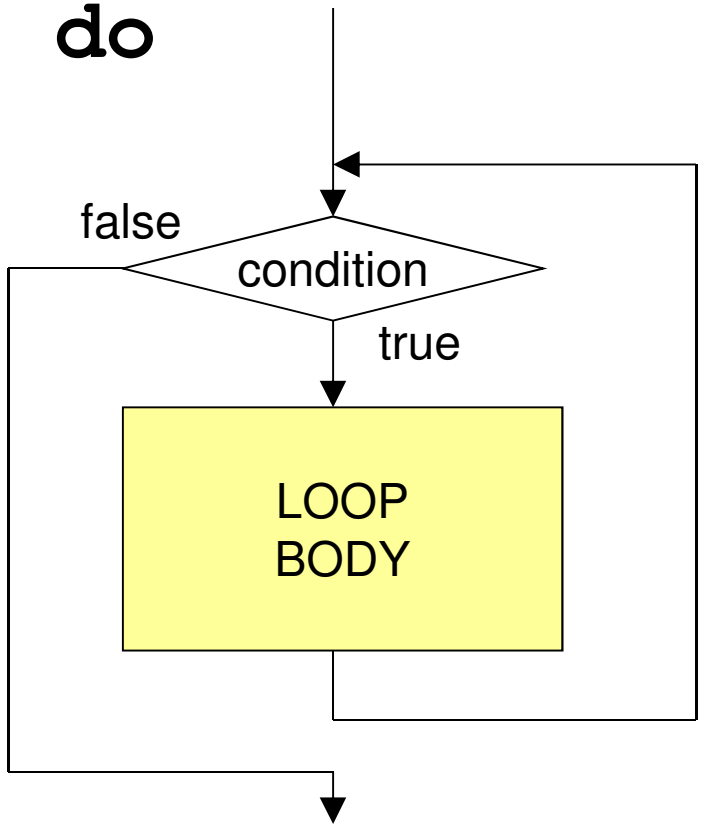
iteration

while loop

Format:

```
while bool_condition do
    loop body
end
```

one or more instructions
to be repeated

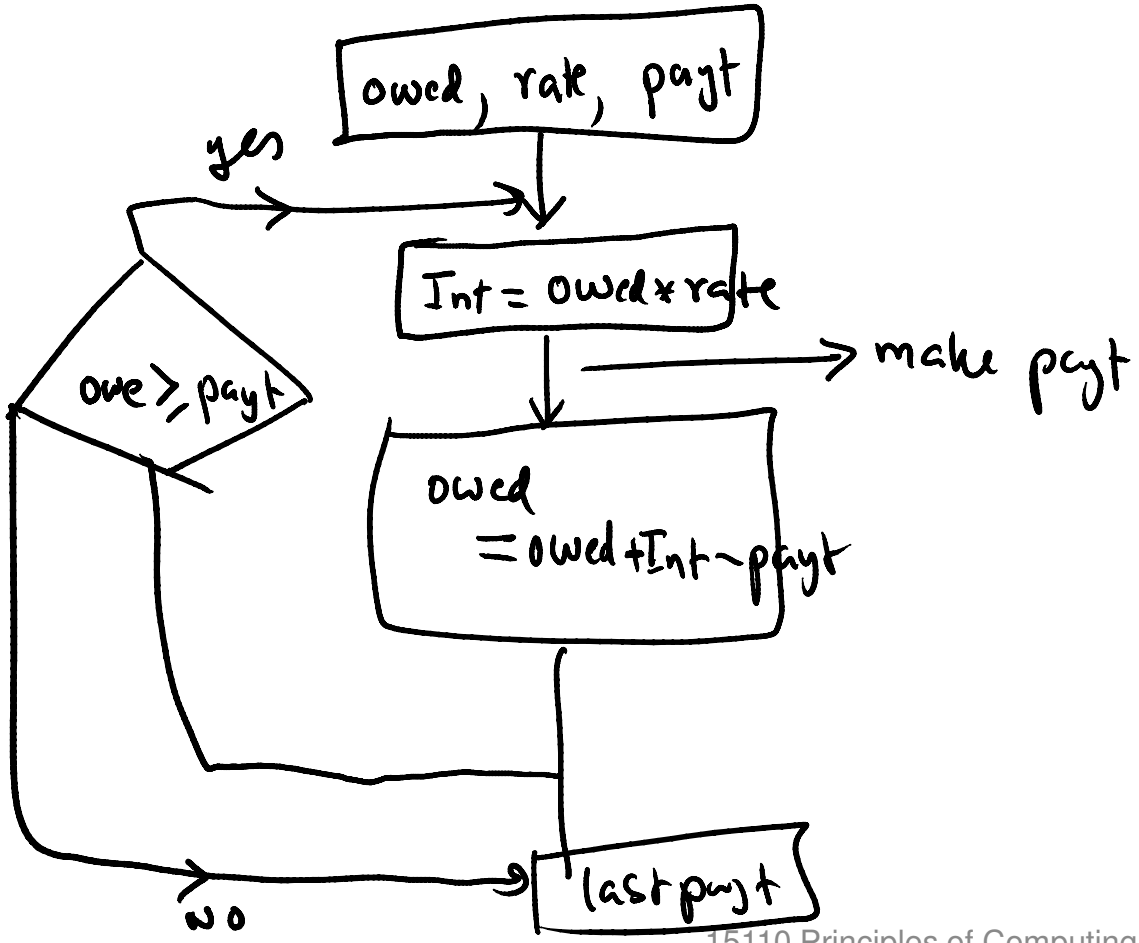


If the loop condition becomes false during the loop body, the loop body still runs to completion before we exit the loop and go on with the next step.

Examples

Interest calculation (flow chart)

- The problem: Given an total amount "owed" and a monthly "rate", find how many "payments" can be deducted from the total. Return the value



Interest calculation (code)

Write a function to check if a number is prime

Use the isPrime function to print all primes between any two numbers

Nested Loops

ASCII ART

- How would you draw a skyscraper?
- How would you combine them to create a skyline?

Next week

- Arrays
- Algorithms on Arrays
- Complexity of algorithms