

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..

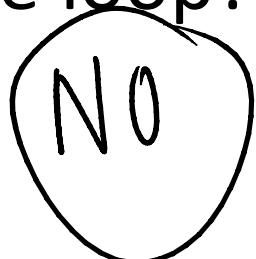
- Can we return more than once from a function?



return x

return y

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



for i in 1..10 do

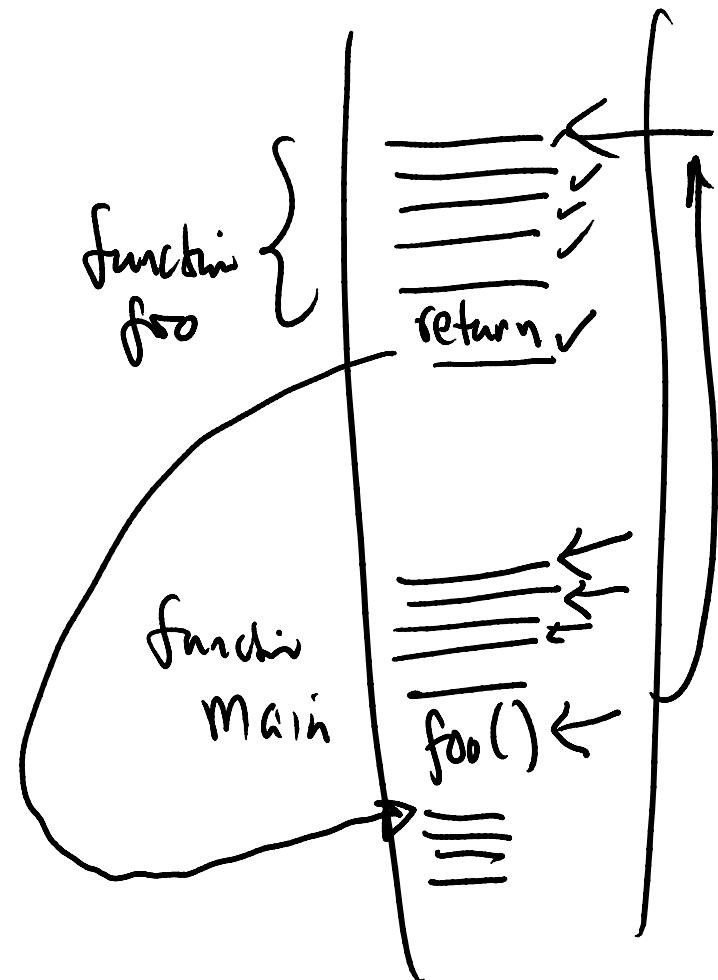
print i
return i

end

Return statement

RAM

- How does it work?



Questions??

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

The diagram illustrates the difference between the `print` and `put` statements in Ruby. It shows two rows of code with arrows indicating the state of the cursor after each statement.

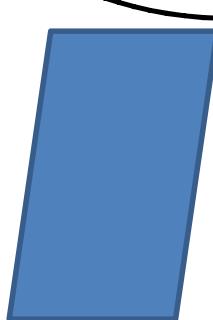
The first row shows the output of `print "5"`. An arrow points from the code to the number 5, with the word "cursor" written above the arrow pointing to the 5.

The second row shows the output of `put "5"`. An arrow points from the code to the string "5\n", with the word "cursor" written below the "\n".

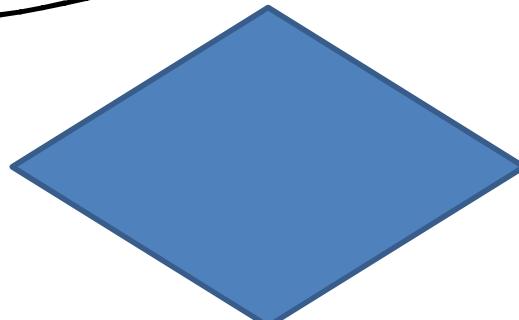
Tools for Developing algorithms

Developing algorithms

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



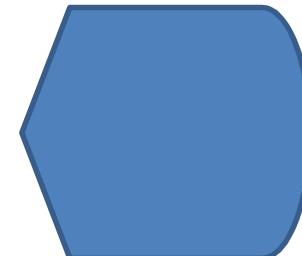
input



decision



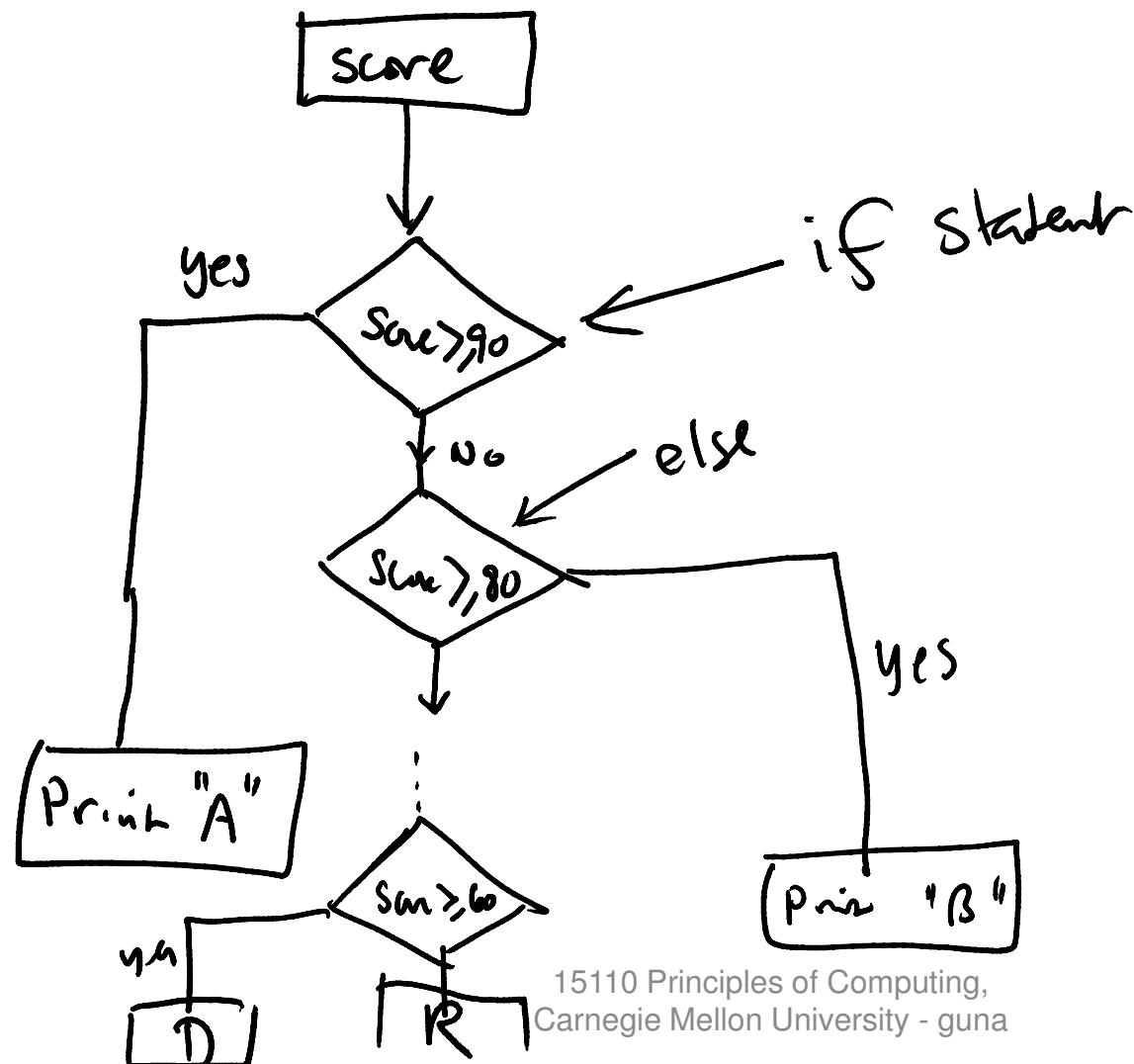
Statement



output

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	F	T

Not	T	F
F	T	F

- DeMorgans Law

$$\neg(A \text{ AND } B) = \neg A \text{ or } \neg B$$

$$\neg(A \text{ or } B) = \neg A \text{ and } \neg B$$

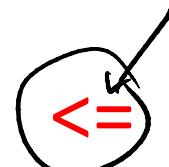
Relational Operators

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

< less than

> greater than

== equal to



less than or equal to

greater than or equal to

not equal to

$$! (x \leq y) \Leftrightarrow x > y \quad ! (\leq) \Leftrightarrow >$$

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
$ = 0$	✓		
$x \geq 1 \text{ or } x < 1$		✓	
$x \geq 1 \text{ and } x < 1$	✓		
$x \geq 1 \text{ or } x < -2$			$x \neq 0$ 30 ✓

→ → →

Branching

if/else statement

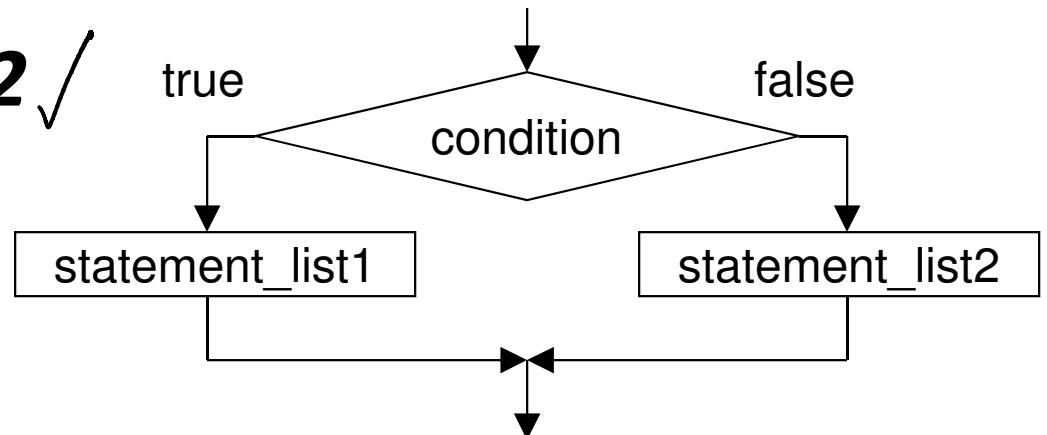
Format:

```
if bool_condition then  
    statement_list1 ✓
```

```
else
```

```
    statement_list2 ✓
```

```
end
```



Write a function to print the grade given a score

```
def grade(score)
    if (score >= 90) then
        print "A"
    else
        if (score >= 80) then
            print "B"
        else
            if (score >= 70) then
                print "C"
            else
                if (score >= 60) then
                    print "D"
                else
                    print "F"
```

Annotations:

- Handwritten circled numbers 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 are placed next to the corresponding levels of nesting in the code.
- Handwritten circled 'or' is placed between the first two 'if' statements.
- Handwritten circled 'end' is placed at the end of the outermost block.

iteration

while loop

Format:

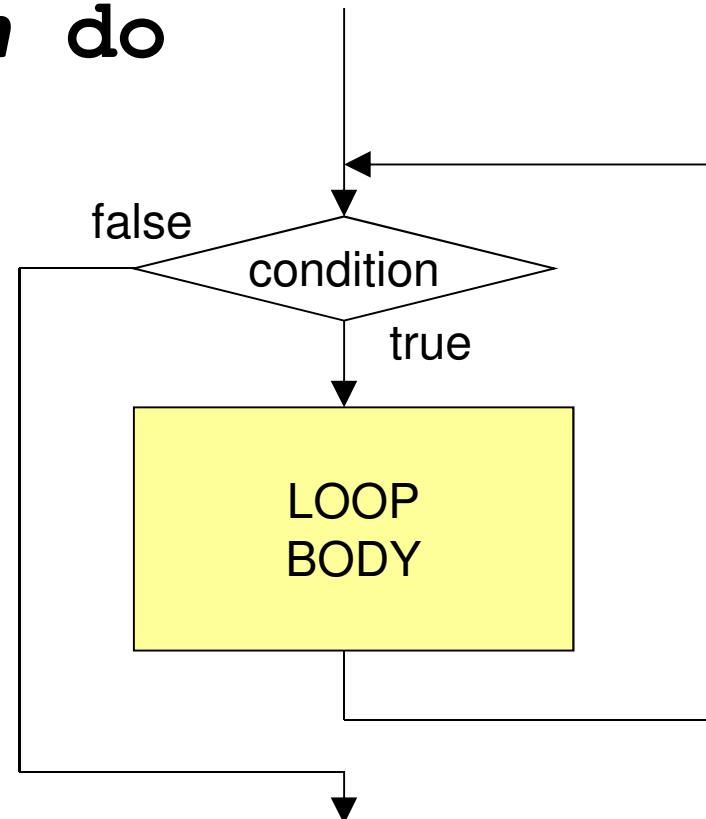
```
while bool_condition do
```

loop body

end

one or more instructions
to be repeated

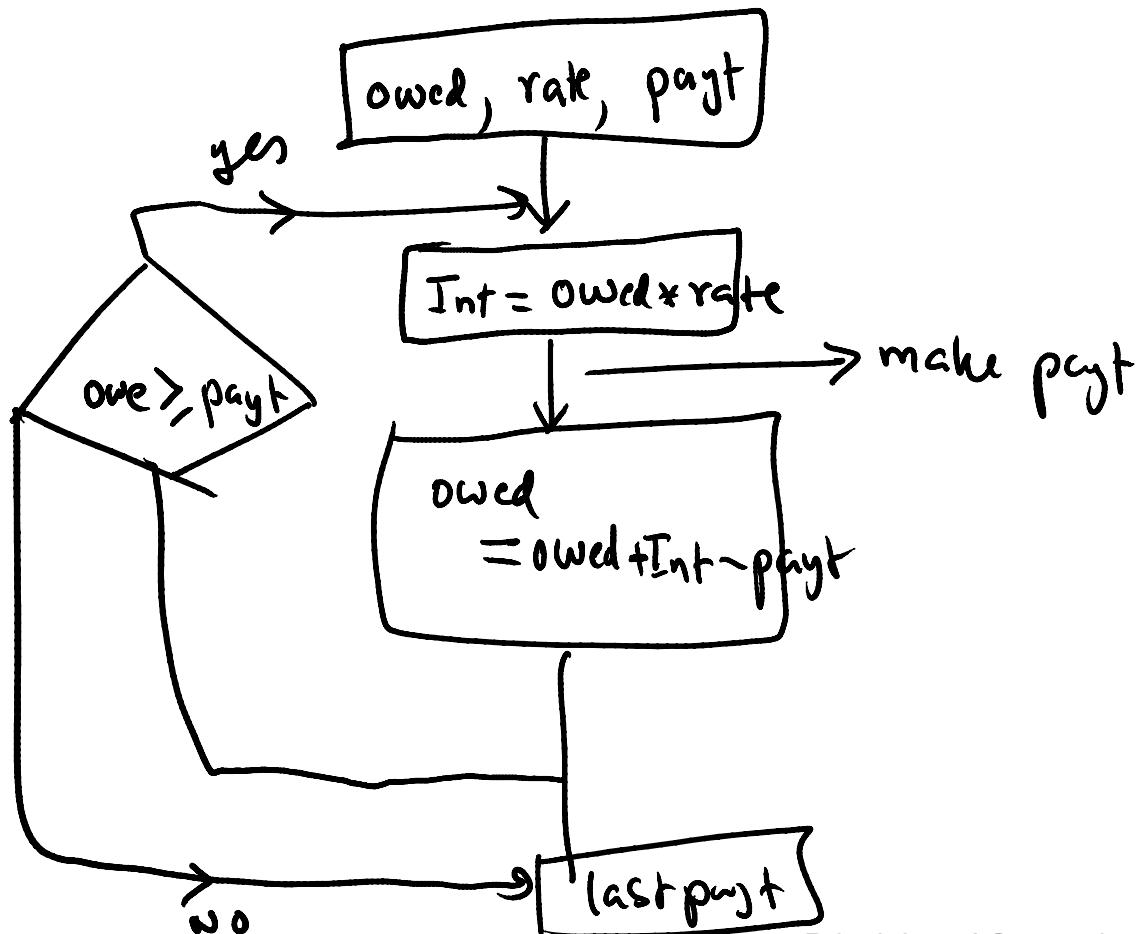
If the loop condition becomes false during
the loop body, the loop body still
runs to completion before we exit
the loop and go on with the next step.



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