



Telling a computer how to behave

(via pseudocode -- a workaround
for Computing's Tower of Babel.)

COS 116: 2/12/2008

Sanjeev Arora

The New York Times

FINDINGS

Hitting It Off, Thanks to Algorithms of Love



Viktor Koen

By JOHN TIERNEY

Jan 29, 2008


Search Romance[Romance help](#)

Pin All Your Romantic Hopes on Google

When you think about it, love is just another search problem. And we've thought about it. A lot. Google Romance™ is our solution.

Google Romance is a place where you can post all types of romantic information and, using our Soulmate Search™, get back search results that could, in theory, include the love of your life. Then we'll send you both on a Contextual Date™, which we'll pay for while delivering to you relevant ads that we and our advertising partners think will help produce the dating results you're looking for.

With Google Romance, you can:

- **Upload your profile** – tell the world who you are, or, more to the point, who you'd like to think you are, or, even more to the point, who you want others to think you are.
- **Search for love** in all (or at least a statistically significant majority of) the right places with Soulmate Search, our eerily effective psychographic matchmaking software.
- **Endure**, via our Contextual Dating option, thematically appropriate multimedia advertising throughout the entirety of your free date.

Learn more: [Take the Tour](#), [Press Release](#), [FAQ](#)

[Take the tour](#)

[Post your Google Romance profile](#)

[Post multiple profiles with a bulk upload file, you sleaze](#)



User A: "Finally I've found my Soul mate! Thanks, Google Romance!"



User B: "I never thought I'd be writing an online dating testimonial. Until I met User A..."

When you do a Soulmate Search, your deeply personal and potentially life-altering search results are produced solely by computer algorithm, without human intervention of any kind.

Note: depending on your personality, you may or may not find this reassuring.



Steps in solving a computational task

- Design an **algorithm**: A precise, unambiguous description for how to compute a solution.
- Express algorithm in **pseudocode**.
- Turn pseudocode into **computer program**.



Discussion Time

Example: Adding two numbers

Imagine you are describing this task to somebody who has never done it. How would you describe it?



Discussion Time

- Our robot is getting ready for a big date...



- How would it identify the cheapest bottle?
(Say it can scan prices)



Solution

- Pick up first bottle, check price
- Walk down aisle. For each bottle, do this:
 - If price on bottle is less than price in hand, exchange for one in hand.



How can we describe an algorithm precisely enough so there is no ambiguity?

Recall: Scribbler's "Language"

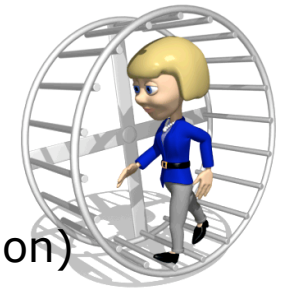
- Several types of simple instructions
 - E.g. "Move forward for 1 s"
- Two types of compound instructions



Loop (2 types)

```
Do 5 times
{
  List of instructions
}
```

```
Do while (condition)
{
  List of instructions
}
```

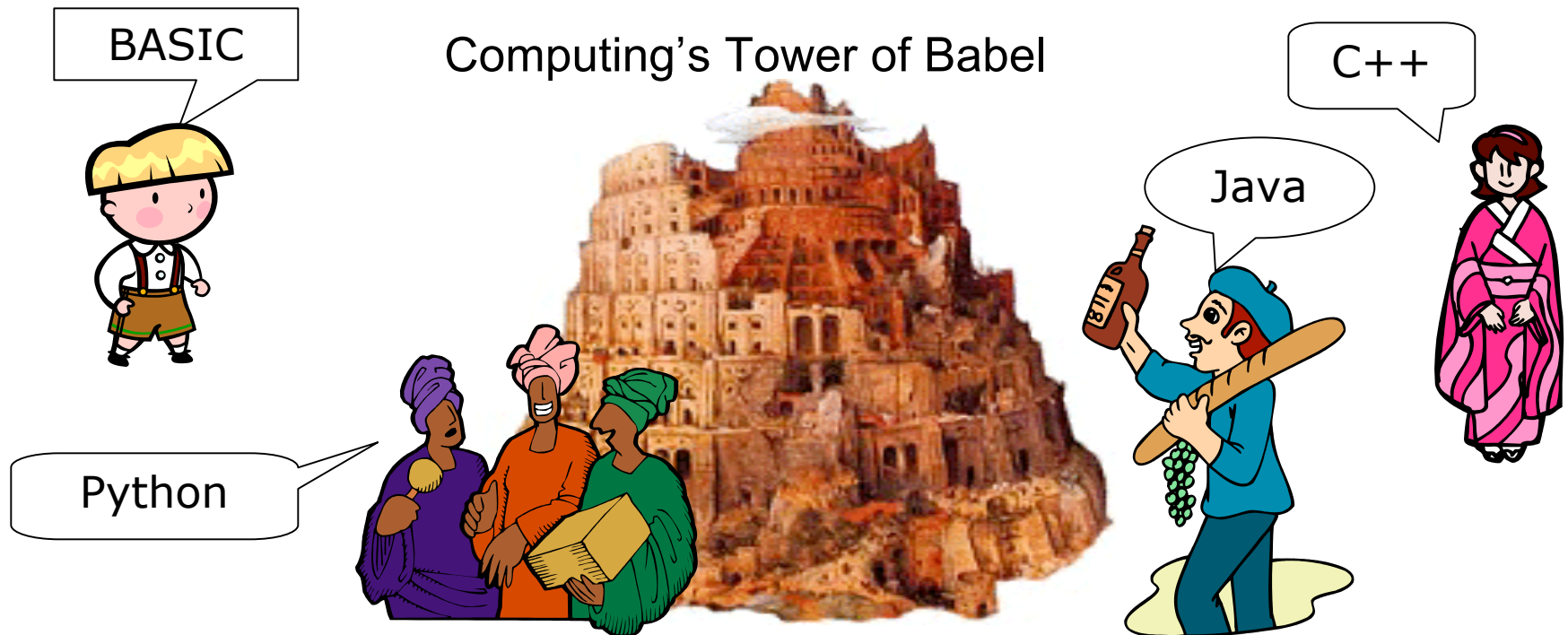


Conditional (a.k.a. Branching)

```
If <condition> Then
{
  List of instructions
}
Else
{
  List of instructions
}
```



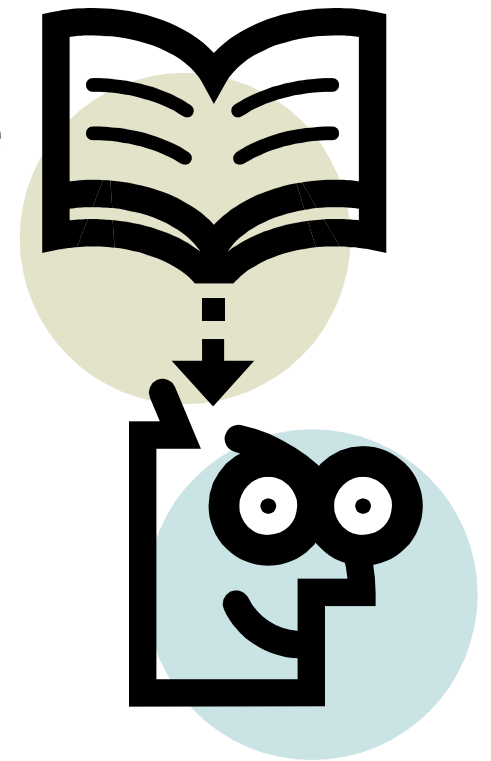
Scribbler language illustrates essential features of all computer languages



- Fundamental features of human languages: nouns/verbs/adjectives, subjects/objects, pronouns, etc.
- Computer languages also share fundamental features, e.g. conditional and loop statements, variables, **ability to perform arithmetic**, etc.

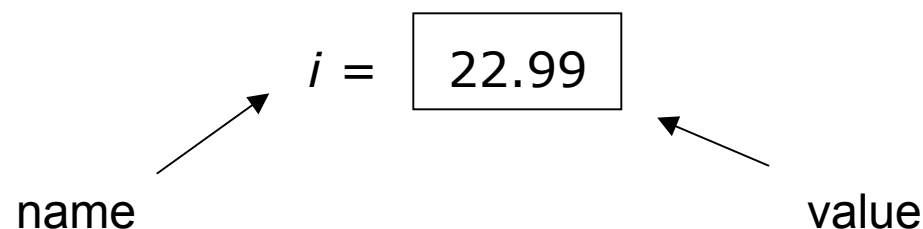
Similar question in different setting

- Robot has n prices stored in memory
- Wants to find minimum price



Computer Memory: simplified view

- A scratchpad that can be perfectly erased and re-written any number of times
- A variable: a piece of memory with a name; stores a “value”





Examples

$$i \leftarrow 5$$

Sets i to value 5

$$j \leftarrow i$$

Sets j to whatever value is in i .
Leaves i unchanged

$$i \leftarrow j + 1$$

Sets i to $j + 1$.
Leaves j unchanged

$$i \leftarrow i + 1$$

Sets i to 1 more than it was.



Arrays

- A is an array of n values, $A[i]$ is the i 'th value

$A =$


40.99	62.99	52.99	...	22.99
-------	-------	-------	-----	-------

- Example: $A[3] = 52.99$



Recall Solution

- Pick up first bottle, check price
- Walk down aisle. For each bottle, do this:
 - If price on bottle is less than price in hand, exchange for one in hand.



Procedure findmin (in pseudocode)

- n items, stored in array A
 - Variables are i , $best$
 - $best \leftarrow 1$
 - Do for $i = 2$ to n
 - {
 - if ($A[i] < A[best]$) then
 - { $best \leftarrow i$ }
- Output $A[best]$.



Another way to do the same

```
best ← 1;  
i ← 1  
Do while (i < n)  
{  
    i ← i + 1;  
    if ( A[ i ] < A[best] ) then  
        { best ← i }  
}
```

```
#include <stdio.h>
int main(void)
{
    int count;

    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");
    return 0;
}
```

NICE TRY.



New problem for robot: sorting



Arrange them so prices increase from left to right.



Solution

Do for $i=1$ to $n-1$

{

Find cheapest bottle among those numbered i to n

Swap that bottle and the i 'th bottle.

}

“selection sort”

Task for Thurs: Write pseudocode for selection sort; due at the start of lecture.



Swapping

- Suppose x and y are variables.
How do you swap their values?
- Need extra variable!

$tmp \leftarrow x$

$x \leftarrow y$

$y \leftarrow tmp$

Aside: History of Algorithm



- Named for Abu Abdullah Muhammad bin Musa al-Khwarizmi (780-850AD)
 - His book "Al-Jabr wa-al-Muqabilah" evolved into today's high school algebra text.
- Notion of algorithm has existed for at least 2000 years (in Hindu, Chinese, and Greek traditions)
- “Variables” in algebra come from the same tradition.



“Findmin? Sorting?? Pseudocode???
How about something more important?”

Fact: This simple pseudocode is all we need to express
all possible computations (topic of a future lecture).

Coming up
on Thurs:

Extreme
Weather

