Telling a computer how to behave

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

COS 116, Spring 2012

Adam Finkelstein

Today’s theme

Google Romance BETA

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

Take the tour

Post your Google Romance profile

Post multiple profiles with a bulk upload file, you sleaze

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.

Algorithm

- A precise unambiguous procedure for accomplishing a task

- Named for Abu Abdullah Muhammad bin Musa al-Khwarizmi
  - His book "Al-Jabr wa-al-Muqabilah" evolved into today's high school algebra text.

- Examples:
  recipe, long division, selection sort.

Discussion
Time

Example: Adding two numbers

Imagine you are describing this task to somebody who has never done it.

How would you describe it?

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

Scribbler

Inputs

Outputs

Scribbler's "Language"

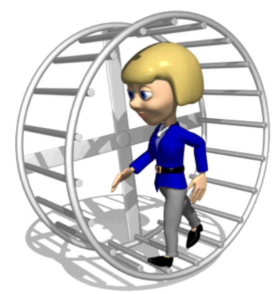
- Several types of simple instructions
  - E.g. "Move forward for 1 s"
- Two types of compound instructions

Conditional

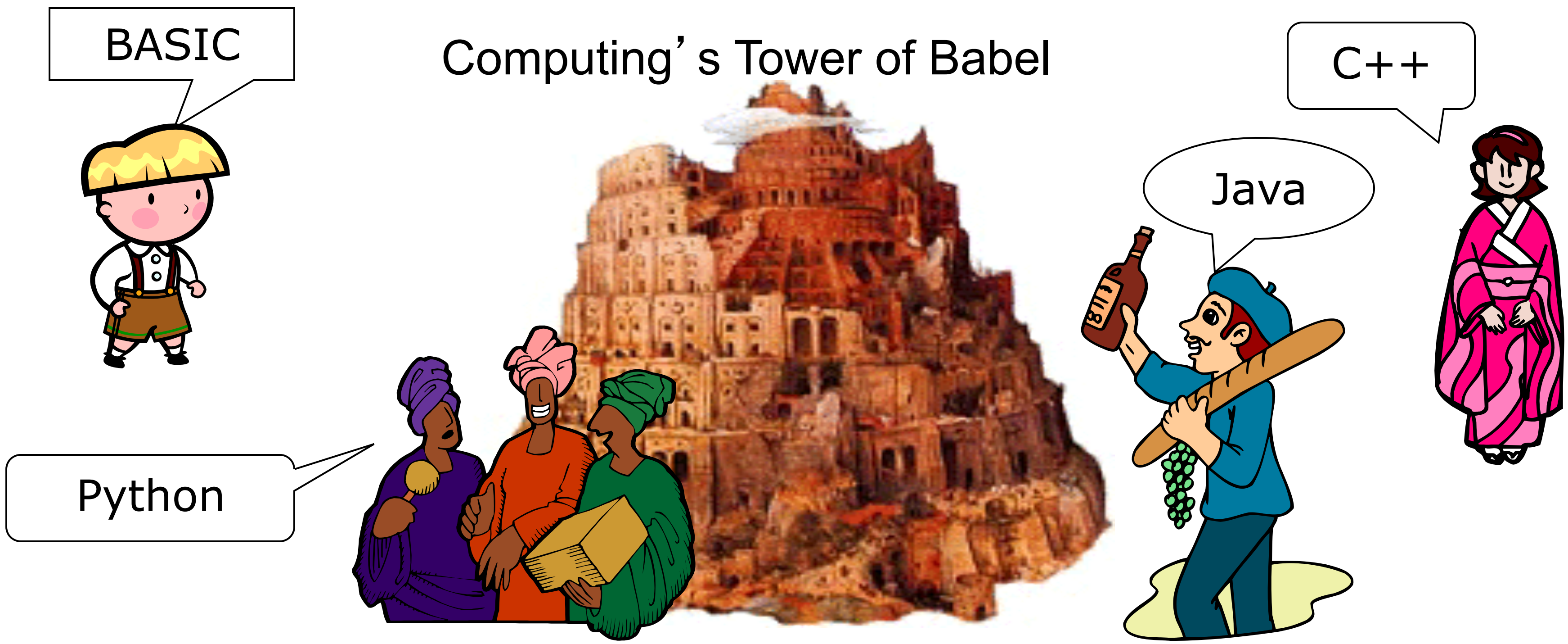
If *condition* Then
{
    List of instructions
}
Else
{
    List of instructions
}

Loop

Do 5 times
{
    List of instructions
}



Scribbler language illustrates essential features of all computer languages

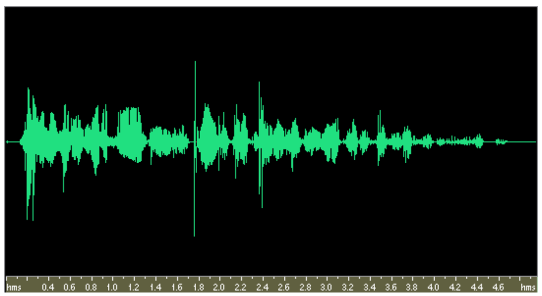


Features of human languages: nouns/verbs/subjects/objects, etc.

Features of computer languages: variables, simple arithmetic instructions, conditional/loop statements

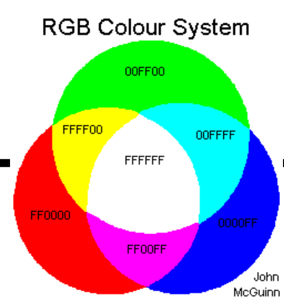
For a computer, everything's a number

Audio waveform



Sequence of Numbers representing frequency, amplitude, etc.

Image



Sequence of Numbers representing color value of each pixel.

General task faced by computer

40.99	62.99	52.99	…	22.99

Given: Sequence of numbers in memory (eg, mp3 song)

Goal: Process these numbers to achieve a desired sequence of numbers as output (eg, electrical signals that drive the headphones and reproduce the song)

A simple problem

- Our robot is getting ready for a big date…

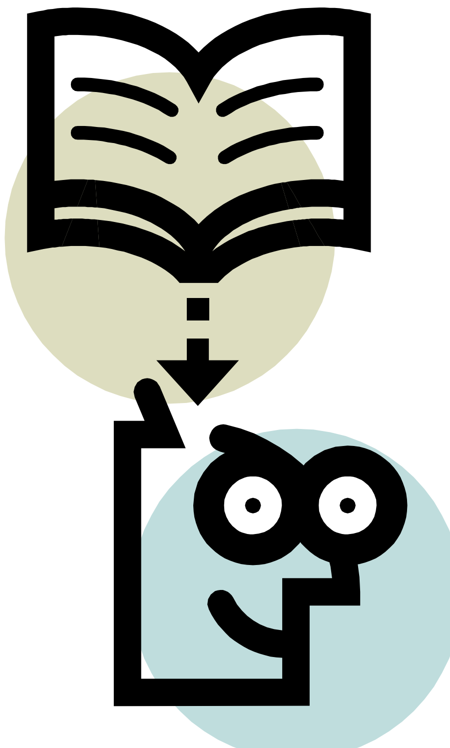
- How would it identify the cheapest bottle? (Say it can pick up a bottle & scan prices.)

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 it with the one in hand.

Similar question in different setting

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



Memory: a 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”

i = $\boxed{22.99}$

name

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

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 it with the one in hand.

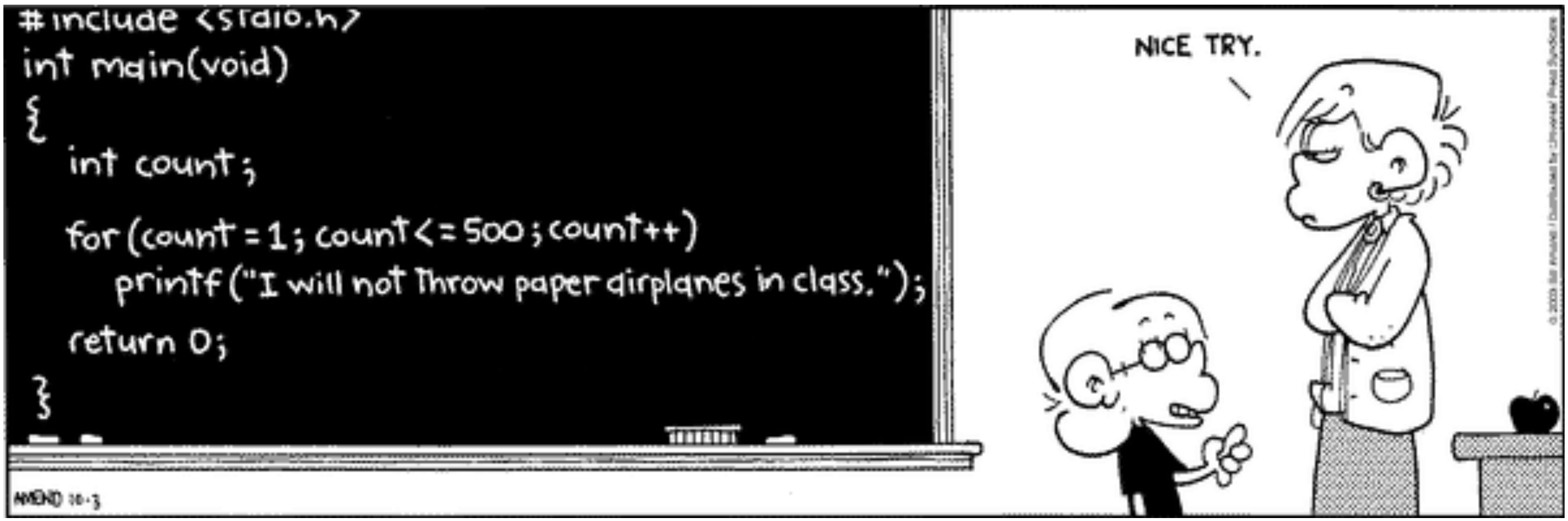
Procedure `findmin`

- *n* items, stored in array *A*
- Variables are *i*, *best*

```
best ← 1
Do for i = 2 to n
  {
     if ( A[i] < A[best] ) then
           best ← i
  }
```

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.
AMEND 10-3

New problem for robot: sorting

Arrange them so prices increase from left to right.

I have to sort n bottles.
Let me find the cheapest bottle and move it to leftmost position. Then I only have to the other bottles to its right.

Solution

Note: we know how to do this!

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"

Swapping

- Suppose *x* and *y* are variables.
  How do you <u>swap</u> their values?

- Need extra variable!

$$tmp \leftarrow x$$
$$x \leftarrow y$$
$$y \leftarrow tmp$$

Love, Marriage & Broken Hearts

Standard disclaimer.

Stable Matching Problem

Problem:

Given *N* men & *N* women, find "suitable" matching

- Everyone lists their preferences from best to worst.

Men's Preference List

Man	1st	2nd	3rd	4th	5th
Victor	Bertha	Amy	Diane	Erika	Clare
Wyatt	Diane	Bertha	Amy	Clare	Erika
Xavier	Bertha	Erika	Clare	Diane	Amy
Yancey	Amy	Diane	Clare	Bertha	Erika
Zeus	Bertha	Diane	Amy	Erika	Clare

best

worst

Stable Matching Problem

<u>Problem</u>:

Given *N* men & *N* women, find “suitable” matching

- Everyone lists their preferences from best to worst.

Women’s Preference List

Woman	1st	2nd	3rd	4th	5th
Amy	Zeus	Victor	Wyatt	Yancey	Xavier
Bertha	Xavier	Wyatt	Yancey	Victor	Zeus
Clare	Wyatt	Xavier	Yancey	Zeus	Victor
Diane	Victor	Zeus	Yancey	Xavier	Wyatt
Erika	Yancey	Wyatt	Zeus	Xavier	Victor

best worst

Stable Matching Problem

- What do we mean by “suitable”?
  - PERFECT: everyone matched monogamously.
  - STABILITY: no incentive for some pair to elope.
    - a pair that is not matched with each other is UNSTABLE if they prefer each other to current partners
    - unstable pair: improve by dumping spouses and eloping
- STABLE MATCHING (Gale and Shapley, 1962) = perfect matching with no unstable pairs.

Example

Men's Preference List

Man	1st	2nd	3rd
Xavier	A	B	C
Yancey	B	A	C
Zeus	A	B	C

Women's Preference List

Woman	1st	2nd	3rd
Amy	Y	X	Z
Bertha	X	Y	Z
Clare	X	Y	Z

- Lavender assignment is a perfect matching. Are there any unstable pairs?

  - Yes. Bertha and Xavier form an unstable pair. They would prefer each other to current partners.

Example

Men's Preference List

Man	1st	2nd	3rd
Xavier	A	B	C
Yancey	B	A	C
Zeus	A	B	C

Women's Preference List

Woman	1st	2nd	3rd
Amy	Y	X	Z
Bertha	X	Y	Z
Clare	X	Y	Z

- Green assignment is a stable matching.

Example

Men's Preference List

Man	1st	2nd	3rd
Xavier	A	B	C
Yancey	B	A	C
Zeus	A	B	C

Women's Preference List

Woman	1st	2nd	3rd
Amy	Y	X	Z
Bertha	X	Y	Z
Clare	X	Y	Z

- Gray assignment is also a stable matching.

Propose-And-Reject Algorithm

- Guarantees a stable matching.

Gale-Shapley Algorithm (men propose)

```
Initialize each person to be free.
while (some man m is free and hasn't proposed to every woman)
{
    w = first woman on m's list to whom he has not yet proposed
    if (w is free)
        assign m and w to be engaged
    else if (w prefers m to her fiancé f)
        assign m and w to be engaged, and f to be free
    else
        w rejects m
}
```

Extensions

- Unacceptable partners
  - Every woman is not willing to marry every man, and vice versa.
  - Some participants declare others as “unacceptable.”
- Sets of unequal size
  - Unequal numbers of men and women, e.g. 100 men & 90 women
- Limited Polygamy
  - e.g., Bill wants to be matched with 3 women.

Matching Residents to Hospitals

- Hospitals ~ Men (limited polygamy allowed).
- Residents ~ Women (more than hospitals)
- Started just after WWII (before computer usage).
- Ides of March, 13,000+ residents are matched.
- Rural hospital dilemma.
  - Certain hospitals (mainly in rural areas) were unpopular and declared unacceptable by many residents.
  - How to find stable matching that benefits rural hospitals?

Homework for next Thurs

(email your answers to pu.cos116@gmail.com by 2/23 noon as part of HW2)

- Write out pseudocode for selection sort.
- Try Gale-Shapley algorithm for previously-shown Amy-Erica / Victor-Zeuss preference lists, but vary the order of choosing man *m*. Does this affect the outcome?
- Try the version where women propose. Does this affect the outcome?
- Bonus question: Try to justify this statement: *When the Gale-Shapley algorithm finishes, there are no unstable pairs.*

Lessons Learned

- Powerful ideas learned in computer science.
- Sometimes deep social ramifications.
  - Hospitals and residents…
  - Historically, men propose to women.
    Why not vice versa?
  - Computer scientists get the best partners!!!

Thursday: the perfect storm…