Citation Key

for more information see: http://open.umich.edu/wiki/CitationPolicy

Use + Share + Adapt

{ Content the copyright holder, author, or law permits you to use, share and adapt. }

PD-GOV **Public Domain – Government**: Works that are produced by the U.S. Government. (17 USC § 105)

PD-EXP **Public Domain – Expired**: Works that are no longer protected due to an expired copyright term.

PD-SELF **Public Domain – Self Dedicated**: Works that a copyright holder has dedicated to the public domain.

CC ZERO **Creative Commons – Zero Waiver**

CC BY **Creative Commons – Attribution License**

CC BY-SA **Creative Commons – Attribution Share Alike License**

CC BY-NC **Creative Commons – Attribution Noncommercial License**

CC BY-NC-SA **Creative Commons – Attribution Noncommercial Share Alike License**

GNU-FDL **GNU – Free Documentation License**

Make Your Own Assessment

{ Content Open.Michigan believes can be used, shared, and adapted because it is ineligible for copyright. }

PD-INEL **Public Domain – Ineligible**: Works that are ineligible for copyright protection in the U.S. (17 USC § 102(b)) *laws in your jurisdiction may differ

{ Content Open.Michigan has used under a Fair Use determination. }

FAIR USE **Fair Use**: Use of works that is determined to be Fair consistent with the U.S. Copyright Act. (17 USC § 107) *laws in your jurisdiction may differ

Our determination **DOES NOT** mean that all uses of this 3rd-party content are Fair Uses and we **DO NOT** guarantee that your use of the content is Fair.

To use this content you should **do your own independent analysis** to determine whether or not your use will be Fair.

Cyberscience: Computational Science and the Rise of the Fourth Paradigm

Please see original comic regarding programming at http://abstrusegoose.com/secret-archives/under-the-hood.

Honors 352, Class #0.8

August E. (Gus) Evrard, PhD

Fall 2010

University of Michigan

Please see original image of a comic on the birth of the ENIAC at http://abstrusegoose.com/17.

today

* read short wikipedia entries on programming

http://en.wikipedia.org/wiki/Computer_programming

http://en.wikipedia.org/wiki/Programming_paradigm#History

* listing of computing languages from UM CIS 400 course

http://groups.engin.umd.umich.edu/CIS/course.des/cis400/

* group projects and membership

1945: the original computer bug!

The First "Computer Bug" Moth found trapped between points at Relay # 70, Panel F, of the Mark II Aiken Relay Calculator while it was being tested at Harvard University, 9 September 1947. The operators affixed the moth to the computer log, with the entry: "First actual case of bug being found". They put out the word that they had "debugged" the machine, thus introducing the term "debugging a computer program". In 1988, the log, with the moth still taped by the entry, was in the Naval Surface Warfare Center Computer Museum at Dahlgren, Virginia, which erroneously dated it 9 September 1945. The Smithsonian Institute's National Museum of American History and other sources have the correct date of 9 September 1947 (Object ID: 1994.0191.01). The Harvard Mark II computer was not complete until the summer of 1947.

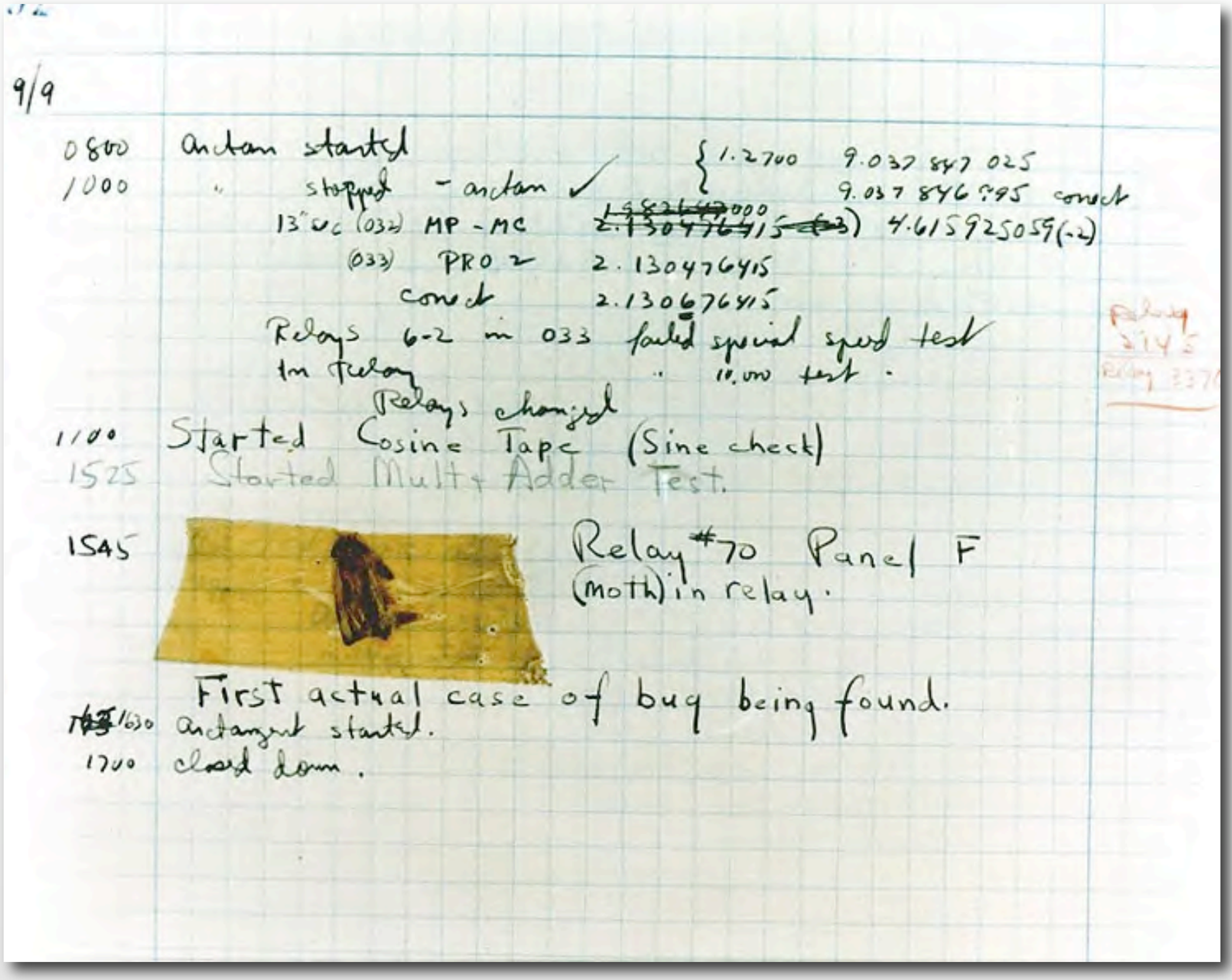
Removed caption read: Photo # NH 96566-KB First Computer "Bug", 1945

9 September 1947(1947-09-09)

U.S. Naval Historical Center Online Library Photograph NH 96566-KN

Courtesy of the Naval Surface Warfare Center, Dahlgren, VA., 1988.

PD-GOV



PD-GOV

tiobe.com language ranking

Ratings

The ratings are calculated by counting hits of the most popular search engines. The search query that is used is

+"<language> programming"

This search query is executed for the top 6 websites of Alexa that meet the following conditions:

* The entry page of the site contains a search facility
* The result of querying the site contains an indication of the number of page hits

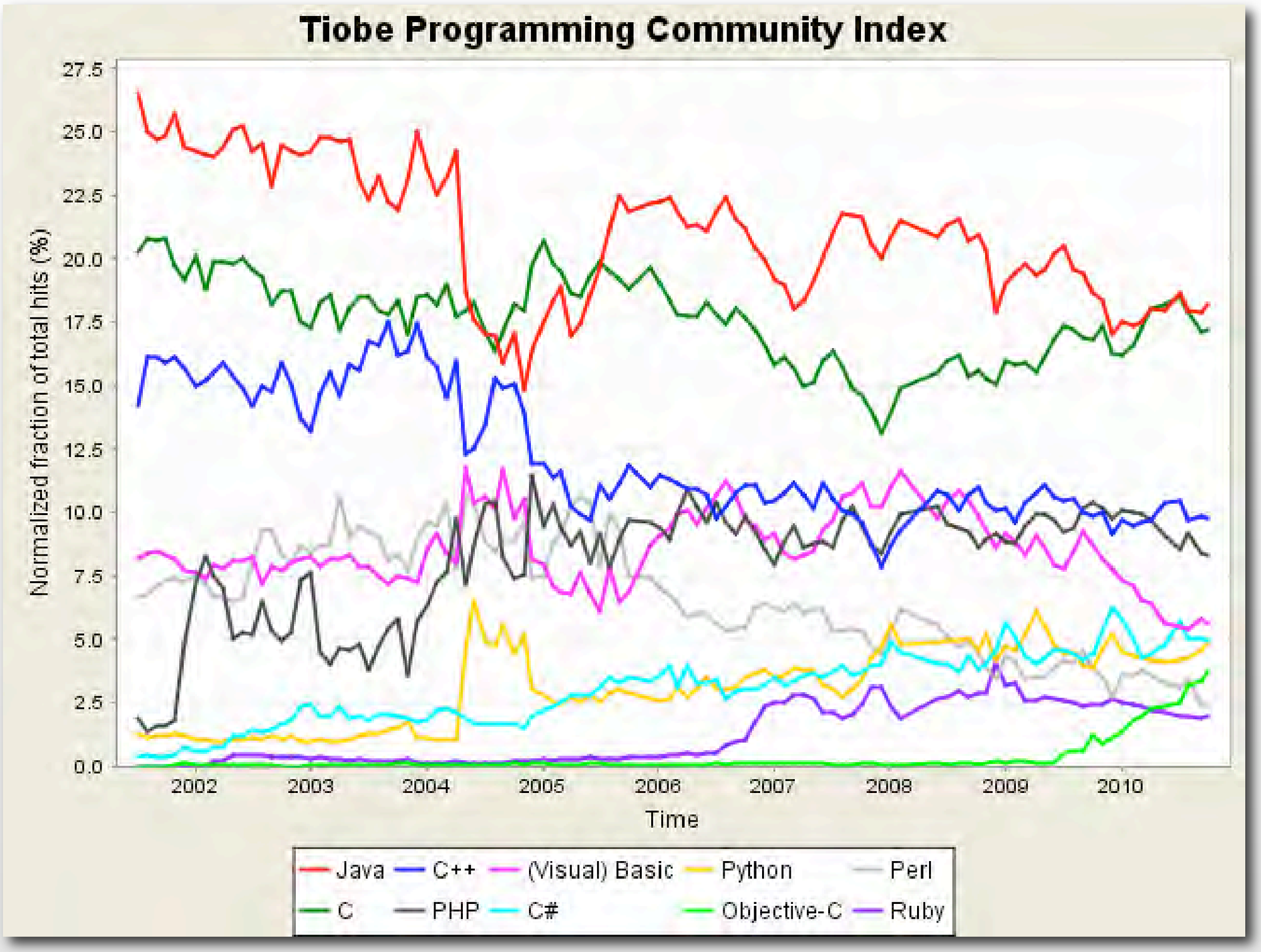
Based on these criteria currently Google, YouTube, Yahoo!, Live, Wikipedia and Blogger are used as search engines. Baidu should be part of this well but the TIOBE index calculator is not capable yet of dealing with Chinese characters. This facility will be added soon.

The number of hits determine the ratings of a language.

Position Oct 2010	Position Oct 2009	Delta in Position	Programming Language	Ratings Oct 2010	Delta Oct 2009	Status
1	1	=	Java	18.166%	-0.48%	A
2	2	=	C	17.177%	+0.33%	A
3	4	↑	C++	9.802%	-0.08%	A
4	3	↓	PHP	8.323%	-2.03%	A
5	5	=	(Visual) Basic	5.650%	-3.04%	A
6	6	=	C#	4.963%	+0.55%	A
7	7	=	Python	4.860%	+0.96%	A
8	12	↑↑↑↑	Objective-C	3.706%	+2.54%	A
9	8	↓	Perl	2.310%	-1.45%	A
10	10	=	Ruby	1.941%	-0.51%	A
11	9	↓↓	JavaScript	1.659%	-1.37%	A
12	11	↓	Delphi	1.558%	-0.58%	A
13	17	↑↑↑↑	Lisp	1.084%	+0.48%	A-
14	24	↑↑↑↑↑↑↑↑↑↑	Transact-SQL	0.820%	+0.42%	A-
15	15	=	Pascal	0.771%	+0.10%	A-
16	18	↑↑	RPG (OS/400)	0.708%	+0.12%	A-
17	29	↑↑↑↑↑↑↑↑↑↑	Ada	0.704%	+0.40%	A--
18	14	↓↓↓↓	SAS	0.664%	-0.14%	B
19	19	=	MATLAB	0.627%	+0.05%	B
20	-	↑↑↑↑↑↑↑↑↑↑	Go	0.626%	+0.63%	B

http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

tiobe.com language ranking



http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

imperative vs. functional styles

Coding styles [edit]

Imperative programs tend to emphasize the series of steps taken by a program in carrying out an action, while functional programs tend to emphasize the composition and arrangement of functions, often without specifying explicit *steps*. A simple example illustrates this with two solutions to the same programming goal (calculating Fibonacci numbers) using the same multi-paradigm language Python.

```
# Fibonacci numbers, imperative style
N=10

first = 0    # seed value fibonacci(0)
second = 1   # seed value fibonacci(1)
fib_number = first + second     # calculate fibonacci(2)
for position in range(N-2):     # iterate N-2 times to give Fibonacci number N (for N > 2)
    first = second              # update the value of the two 'previous' variables
    second = fib_number
    fib_number = first + second # update the result value to fibonacci(position)
print fib_number
```

A functional version has a different feel to it:

```
# Fibonacci numbers, functional style
def fibonacci(N):  # Fibonacci number N (for N >= 0)
    if N <= 1: return N    # base cases
    else: return fibonacci(N-1) + fibonacci(N-2)   # recursively calculate fibonacci(N)

print fibonacci(10)
```

The imperative style describes the intermediate steps involved in calculating `fibonacci(N)`, and places those steps inside a loop statement. In contrast, the functional style describes the mathematical equation that defines a `fibonacci(N)` number with respect previous numbers in the Fibonacci sequence, where intermediate calculation steps are calculated using recursion.

Additional Source Information

for more information see: http://open.umich.edu/wiki/CitationPolicy

Slide 3: Please see original comic regarding programming at http://abstrusegoose.com/secret-archives/under-the-hood.

Slide 4: Please see original image of a comic on the birth of the ENIAC at http://abstrusegoose.com/17.

Slide 6, Quote (left): United States Navy
Slide 6, Image (right): United States Navy

Slide 7: “TIOBE Programming Community Index for September 2011,” Tiobe, http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Slide 8: “TIOBE Programming Community Index for September 2011,” Tiobe, http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html

Slide 9: Wikipedia, http://en.wikipedia.org/wiki/Functional_programming, CC: BY-SA 3.0, http://creativecommons.org/licenses/by-sa/3.0/.