Introduction

CMPT 110

Questions

- What is your major, please.
- Why are you taking this course?
- What do you expect to learn?

CMPT 110

- This course is intended for students with little or no computing background.

Prerequisites

BC Mathematics 12 (or equivalent) or any 100 level MATH course. Students who have obtained credit for, or are currently enrolled in a computing science course at the 200 level or higher, or ITEC 240, 241 or 242 may not take CMPT 110 for further credit except with permission of the School of Computing Science. Quantitative.

CMPT 110

- This course is intended for students with little or no computing background.
- This is both a *theory* and *applied* course.

Preamble

What is Computing Science?

What is Computing Science?

computing science

Web definitions

Computer Science is the scientific and practical approach to computation and its applications. It is the systematic study of the feasibility, structure, expression, and mechanization of the methodical processes that underlie the acquisition, representation, processing, storage, communication of, ...
http://en.wikipedia.org/wiki/Computing_Science

What is Computing Science?

computing science

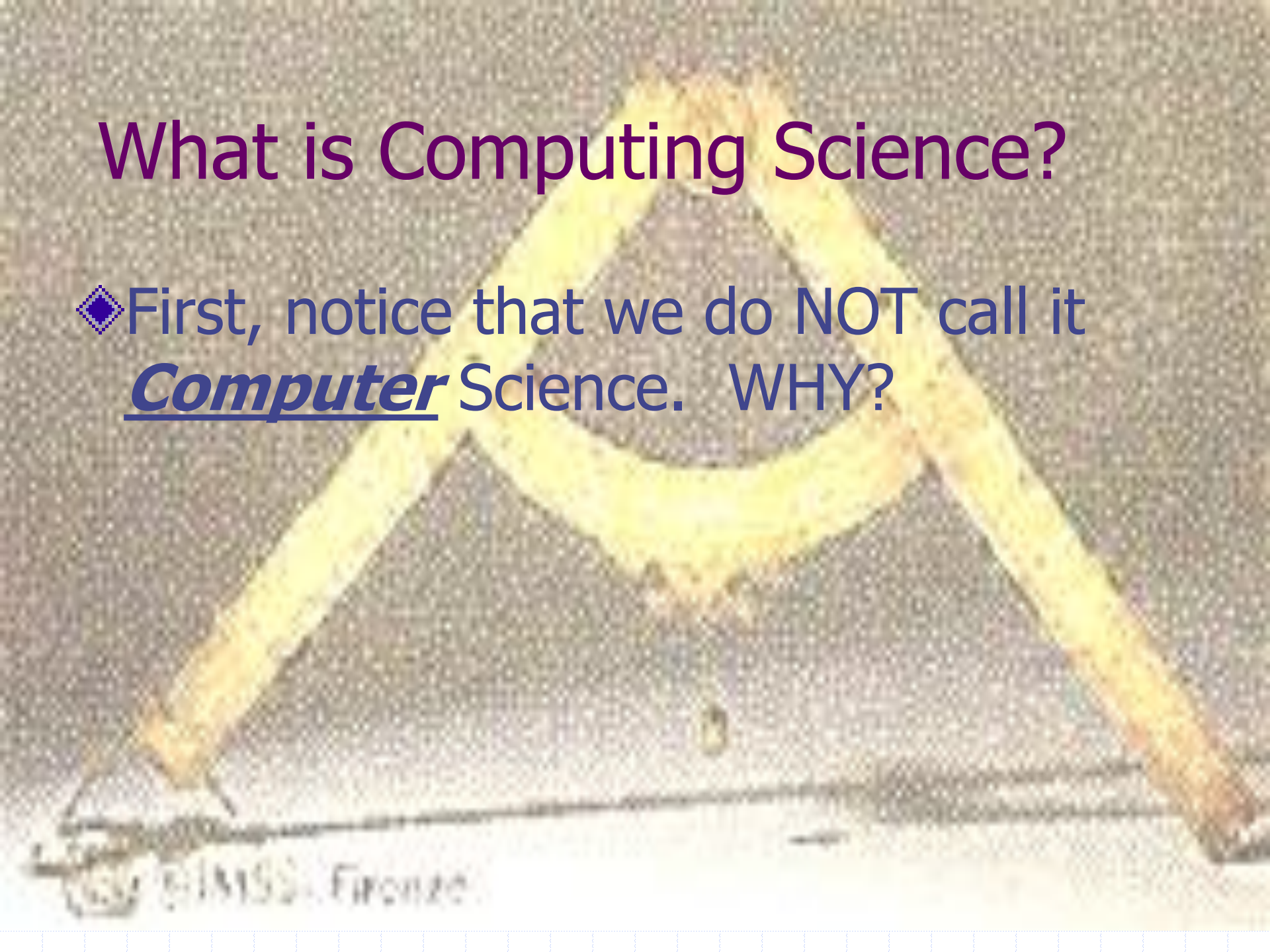
Web definitions

Computer Science is the scientific and practical approach to computation and its applications. It is the systematic study of the feasibility, structure, expression, and mechanization of the methodical processes that underlie the acquisition, representation, processing, storage, communication of, ...
http://en.wikipedia.org/wiki/Computing_Science

Not the best definition…

What is Computing Science?

- First, notice that we do NOT call it ***Computer*** Science. WHY?



What is Computing Science?

- First, notice that we do NOT call it ***Computer*** Science. WHY?
- Think of the most fundamental concept related to computing anything.

DEFINITION: Computing Science

The study of *algorithms*,

DEFINITION: Computing Science

The study of *algorithms*, including

- Their formal and mathematical properties,

DEFINITION: Computing Science

The study of *algorithms*, including

- Their formal and mathematical properties,
- Their hardware realizations,

DEFINITION: Computing Science

The study of *algorithms*, including

- Their formal and mathematical properties,
- Their hardware realizations,
- Their linguistic realizations,

DEFINITION: Computing Science

The study of *algorithms*, including

- Their formal and mathematical properties,
- Their hardware realizations,
- Their linguistic realizations,
- Their applications.

- Schneider and Gersting

Questions

- What is an algorithm?
- Is a recipe an algorithm?
- Is there an algorithm that can write other algorithms?
- Is there a code of ethics associated with the construction of algorithms?

Definition: Algorithm

A well-ordered collection of unambiguous and effectively computable operations that, when executed, produces a result and halts in a finite amount of time [within the context of solving a problem].

- Schneider and Gersting

Definition: Algorithm

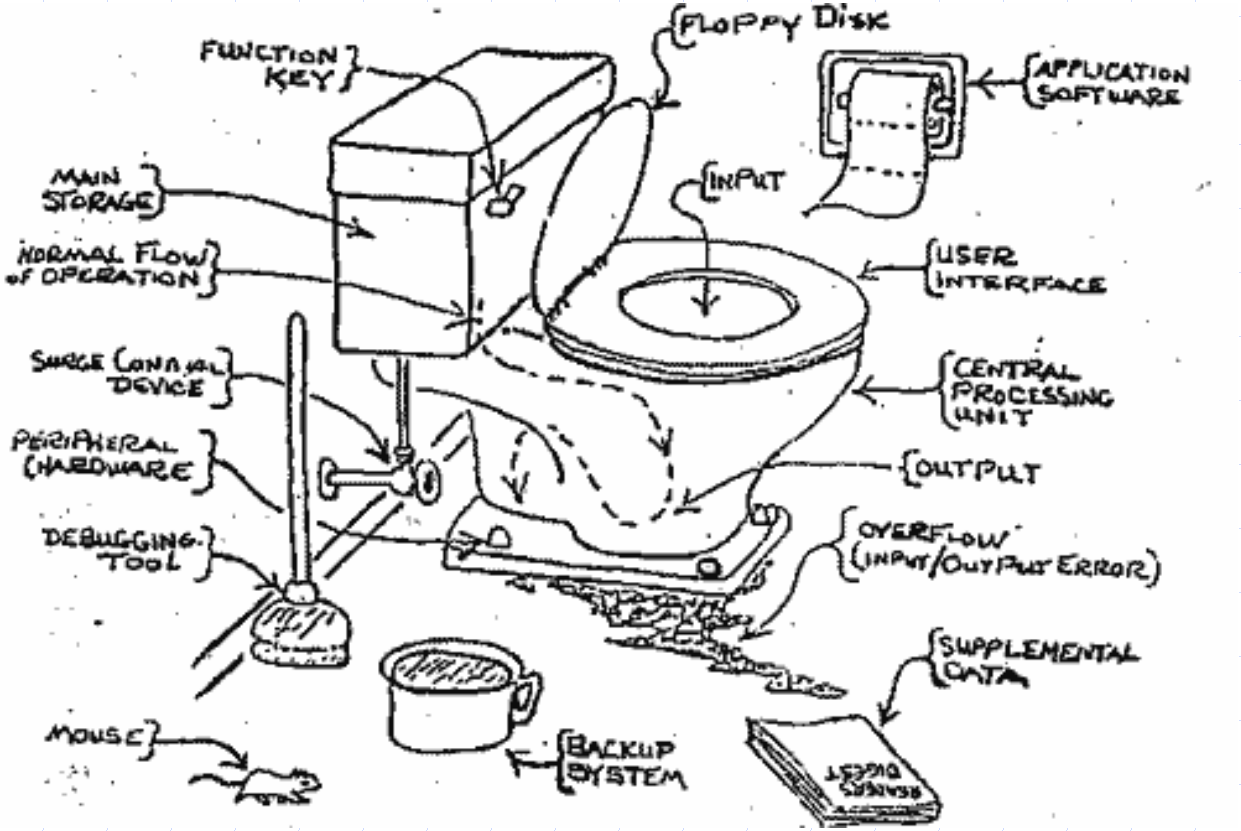
A well-ordered collection of unambiguous and effectively computable operations that, when executed, produces a result and halts in a finite amount of time [within the context of solving a problem].

- Schneider and Gersting

Note: An algorithm is a **formal** structure. Computer programs (S/W) are the implementations of algorithms.

Preliminary Question:

What kind of machine do we want in order to implement algorithms?



Basic Characteristics of a "Computer

- It will take input and produce a desired output (I/O) according to some recipe.

Basic Characteristics of a "Computer

- It will take input and produce a desired output (I/O) according to some recipe.
- Put another way, it will execute algorithms.

Basic Characteristics of a "Computer

- It will take input and produce a desired output (I/O) according to some recipe.
- Put another way, it will execute algorithms.
- That is, it will store programs and data, execute the steps sequentially, and produce an output for an associated input.

Basic Characteristics of a "Computer

- It will take input and produce a desired output (I/O) according to some recipe.
- Put another way, it will execute algorithms.
- That is, it will store programs and data, execute the steps sequentially, and produce an output for an associated input.
- That means, it will require memory, a "processor," the ability to perform basic functions (like arithmetic/logic), etc.

Basis of Hardware

DEFINITION: Von Neumann Architecture

The Von Neumann architecture is a model for designing and building computers that is based on the following three characteristics:

Basis of Hardware

DEFINITION: Von Neumann Architecture

The Von Neumann architecture is a model for designing and building computers that is based on the following three characteristics:

1. Four major subsystems called MEMORY, Input/Output (I/O), Arithmetic/Logic Unit (ALU), and the Control Unit (CU).

Basis of Hardware

DEFINITION: Von Neumann Architecture

The Von Neumann architecture is a model for designing and building computers that is based on the following three characteristics:

1. Four major subsystems called MEMORY, Input/Output (I/O), Arithmetic/Logic Unit (ALU), and the Control Unit (CU).
2. The STORED PROGRAM CONCEPT

Basis of Hardware

DEFINITION: Von Neumann Architecture

The Von Neumann architecture is a model for designing and building computers that is based on the following three characteristics:

1. Four major subsystems called MEMORY, Input/Output (I/O), Arithmetic/Logic Unit (ALU), and the Control Unit (CU).
2. The STORED PROGRAM CONCEPT
3. SEQUENTIAL EXECUTION OF INSTRUCTIONS

This is the basis of MACROARCHITECTURE.

Basis of Hardware

DEFINITION: Von Neumann Architecture

The Von Neumann architecture is a model for designing and building computers that is based on the following three characteristics:

1. Four major subsystems called MEMORY, Input/Output (I/O), Arithmetic/Logic Unit (ALU), and the Control Unit (CU).
2. The STORED PROGRAM CONCEPT
3. SEQUENTIAL EXECUTION OF INSTRUCTIONS

This is the basis of MACROARCHITECTURE.

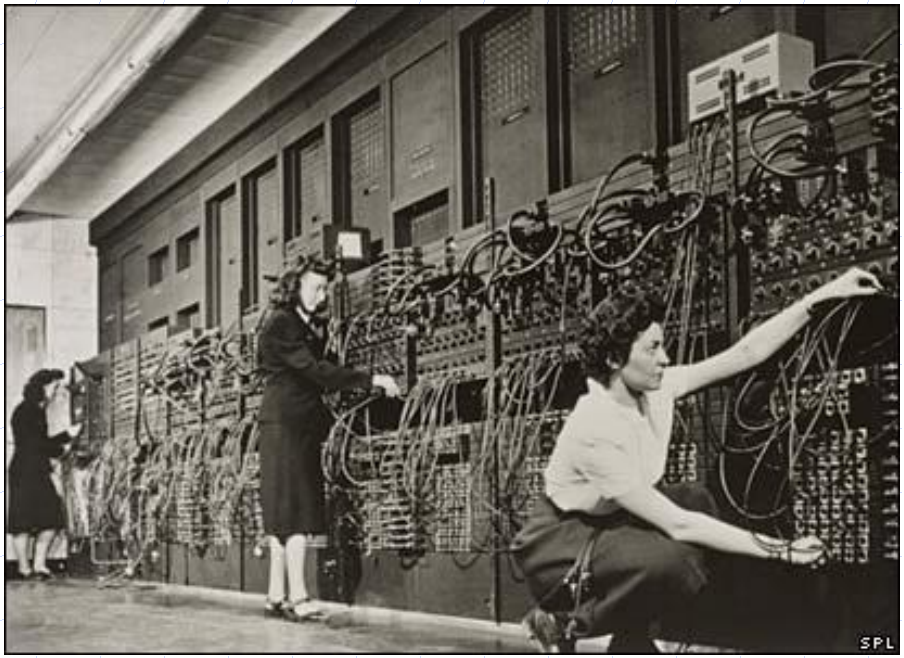
Let us now turn to the design of Von Neumann Machines

Historic Development

Historic Development

- The first electronic computers were built in the 1940s.

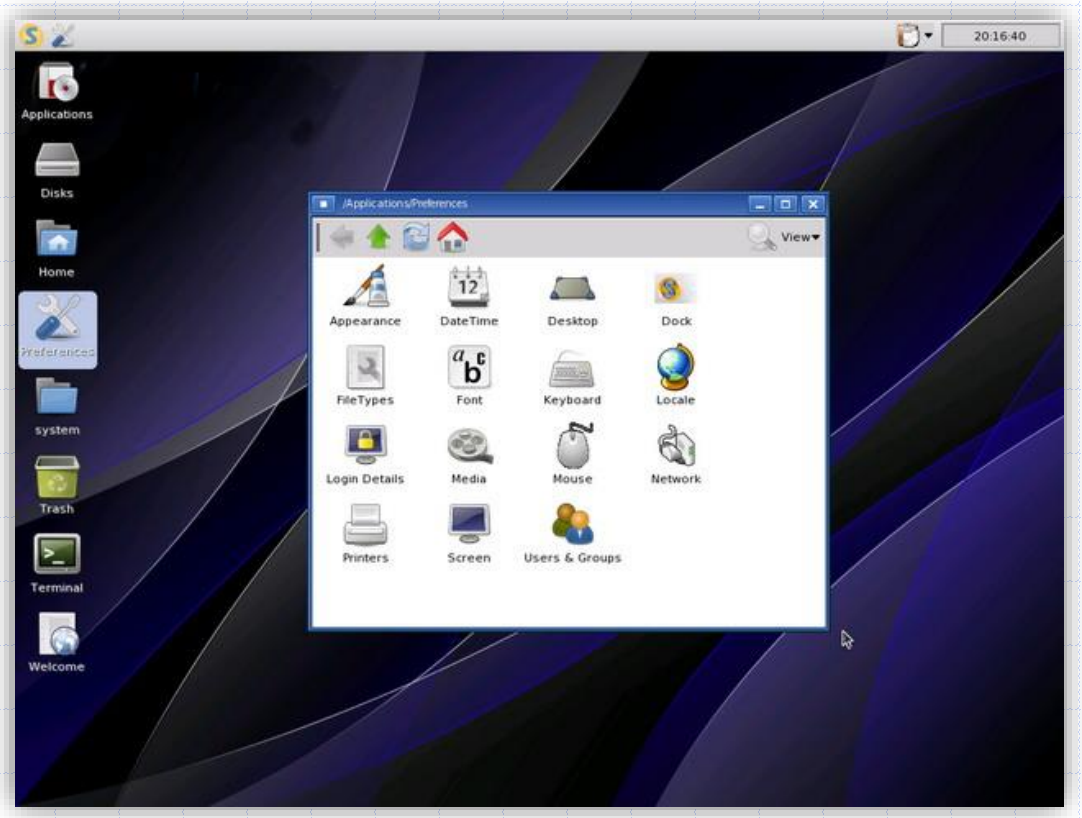
Machine-centered



Historic Development

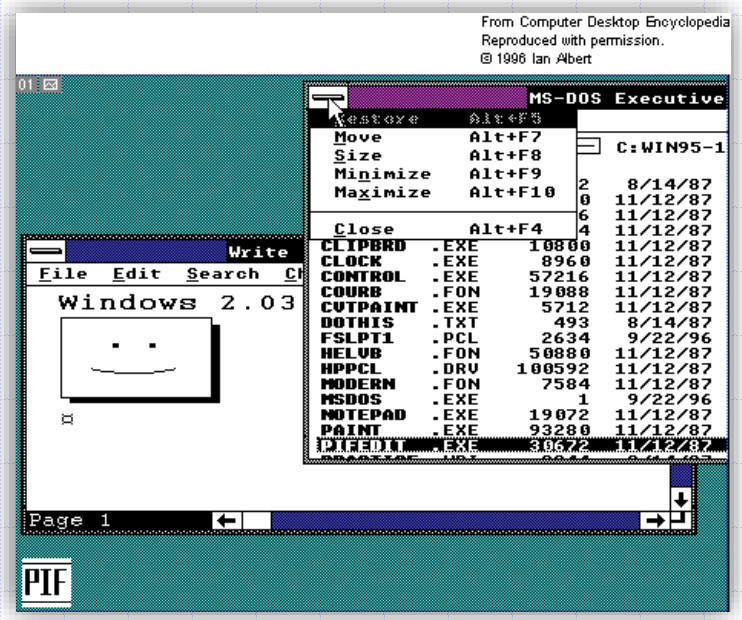
- Today…

User-centered



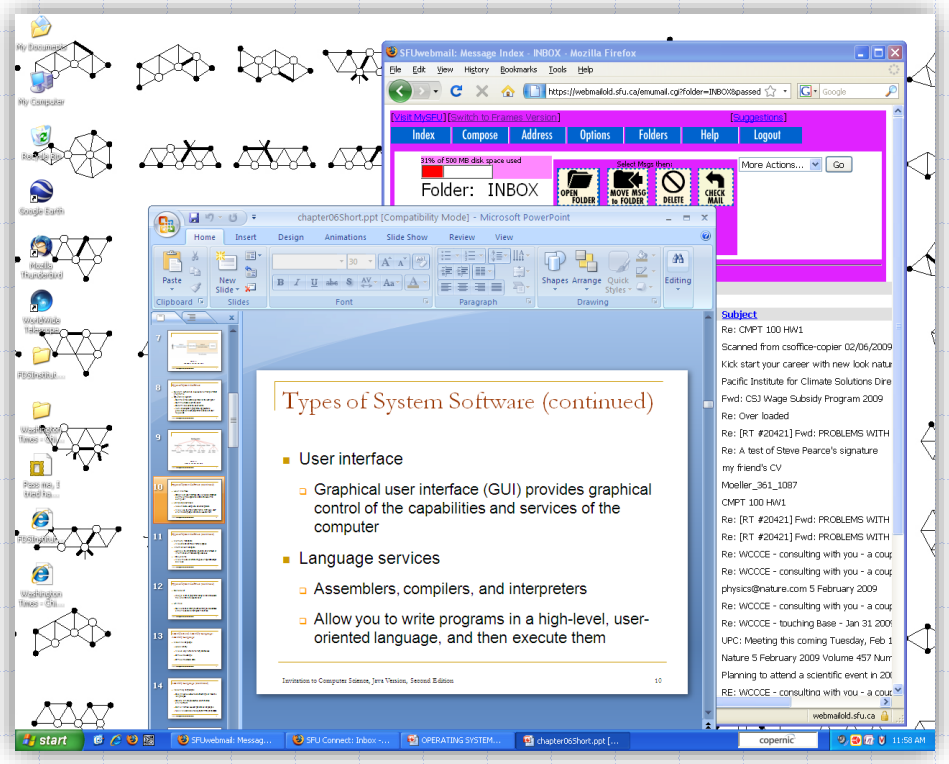
Historic Development

- THE TURNING POINT IS OFTEN REGARDED TO BE ASSOCIATED WITH THE MASSIVE ECONOMIC EXPANSION OF THE COMMERCIAL COMPUTER MARKET ASSOCIATED WITH WINDOWS (from command to WIMP).



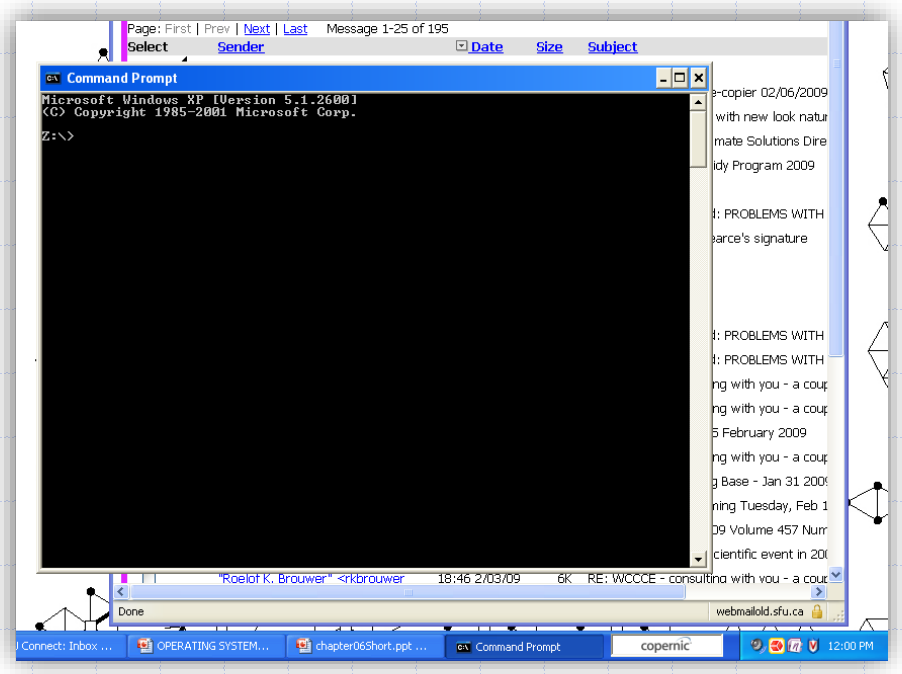
Historic development – User Interface

- GUI – Graphic User Interface
  - WIMP (Windows, Icons, Menus, Pointer), EASY



Historic development – User Interface

- GUI – Graphic User Interface
  - WIMP (Windows, Icons, Menus, Pointer), EASY
- CLI – Command Line Interface
  - Typed text
  - *I.e.,* DOS



Historic development – User Interface

- GUI – Graphic User Interface
  - WIMP (Windows, Icons, Menus, Pointer), EASY
- CLI – Command Line Interface
  - Typed text
  - *I.e.,* DOS
- HYBRIDS – GUI + CLI
  - *i.e.,* UNIX

Historic development – User Interface

- GUI – Graphic User Interface
  - WIMP (Windows, Icons, Menus, Pointer), EASY
- CLI – Command Line Interface
  - Typed text
  - *I.e.,* DOS
- HYBRIDS – GUI + CLI
  - *i.e.,* UNIX
- The evolution has been from "machine-centered" to "user-centered".

Visual Basic in Historic Perspective

Visual basic programming language allows programmers to create software interface and codes in an easy to use graphical environment. VB is the combination of different components that are used on forms having specific attributes and actions with the help of those components. Jan 25, 2015

Visual Basic - The Importance of Visual Basic programming Language ...
https://www.linkedin.com/pulse/visual-basic-importance-programming-language-fareed

Visual Basic in Historic Perspective

Visual basic programming language allows programmers to create software interface and codes in an easy to use graphical environment. VB is the combination of different components that are used on forms having specific attributes and actions with the help of those components. Jan 25, 2015

Visual Basic - The Importance of Visual Basic programming Language ...
https://www.linkedin.com/pulse/visual-basic-importance-programming-language-fareed

Key words you should learn ASAP:
OOP (object-oriented programming), Event-Driven, Graphical Environment, IDE (Integrated Development Environment), 3GL (3rd Generation Language).

Question Revisited

- What is a Programming Language?

Question Revisited

- What is a Programming Language?

A **programming language** is a formal language that specifies a set of instructions that can be used to produce various kinds of output. Programming languages generally consist of instructions for a computer. Programming languages can be used to create programs that implement specific algorithms.

WIKIPEDIA
The Free Encyclopedia

Question Revisited

- What is a Programming Language?

A **programming language** is a formal language that specifies a set of instructions that can be used to produce various kinds of output. Programming languages generally consist of instructions for a computer. Programming languages can be used to create programs that implement specific algorithms.

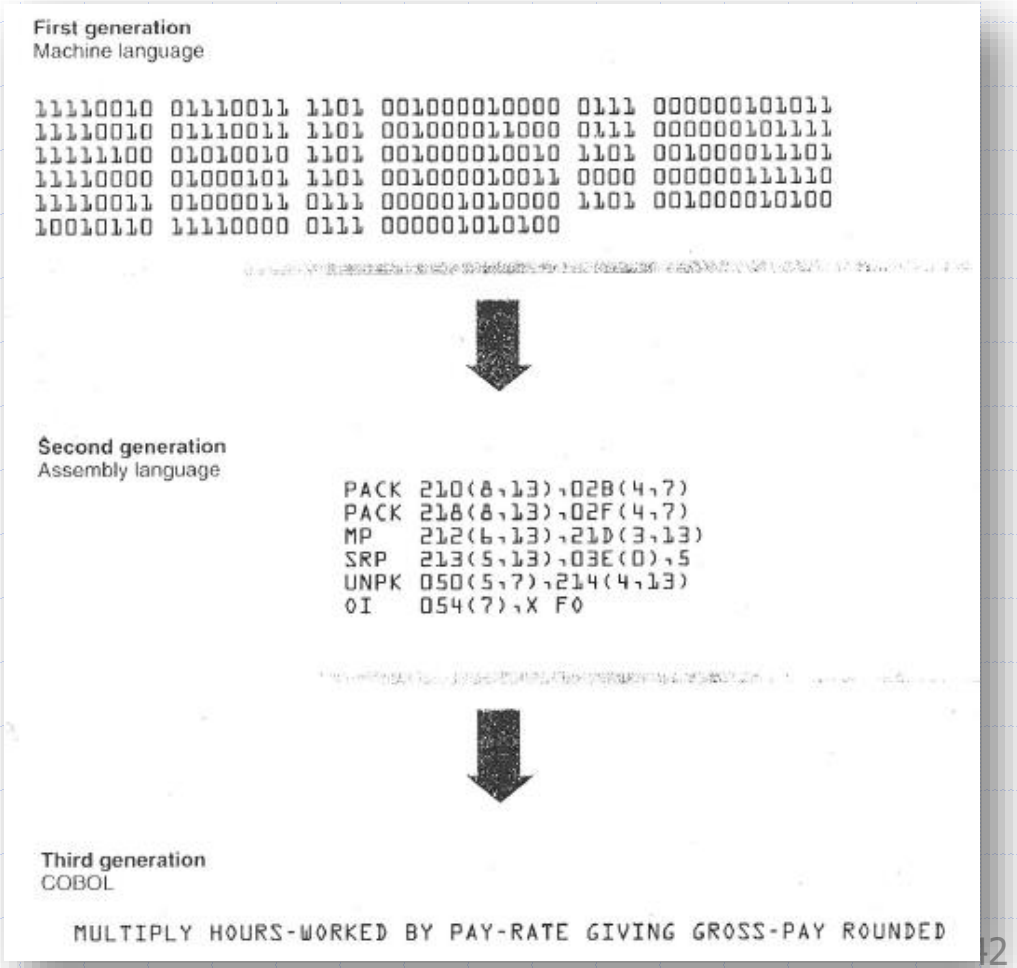
- Programming languages have evolved in step with the evolution of H/W.

WIKIPEDIA
The Free Encyclopedia

The Evolution of Programming Languages

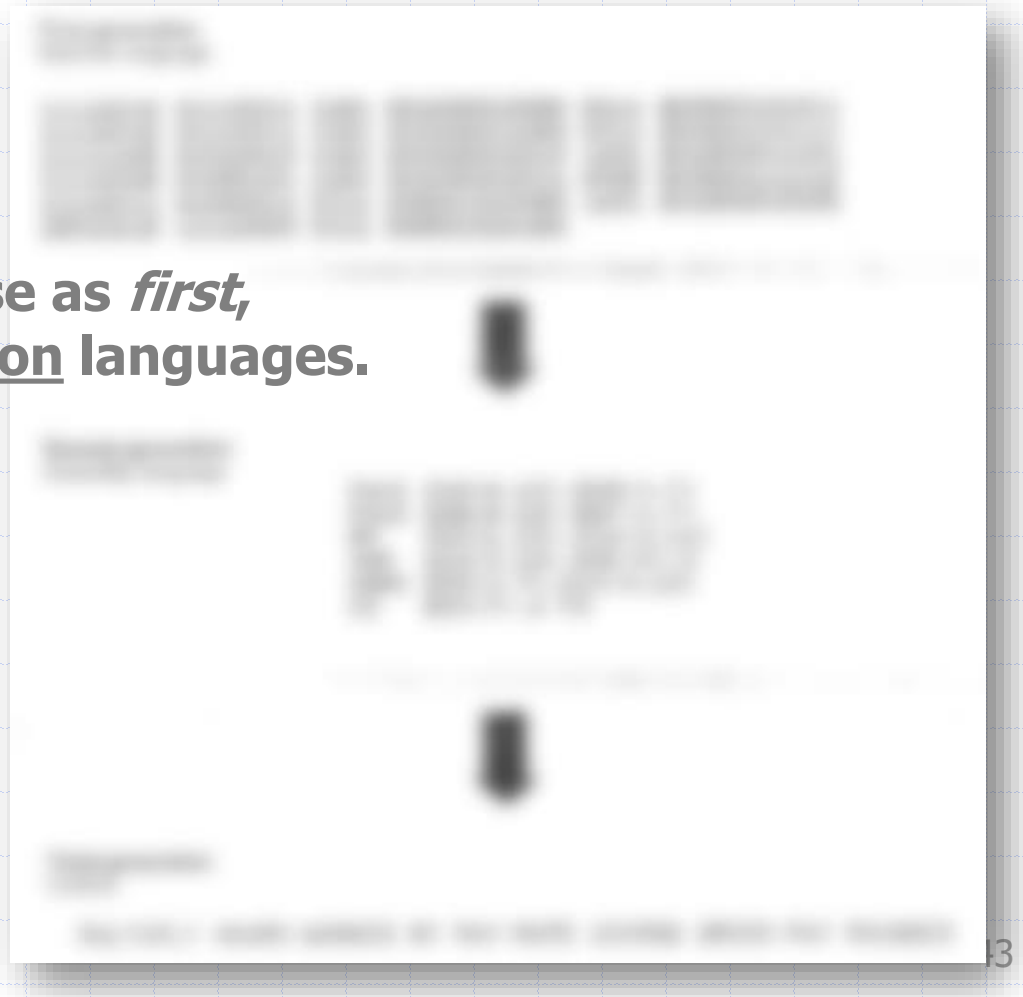
NOT user friendly

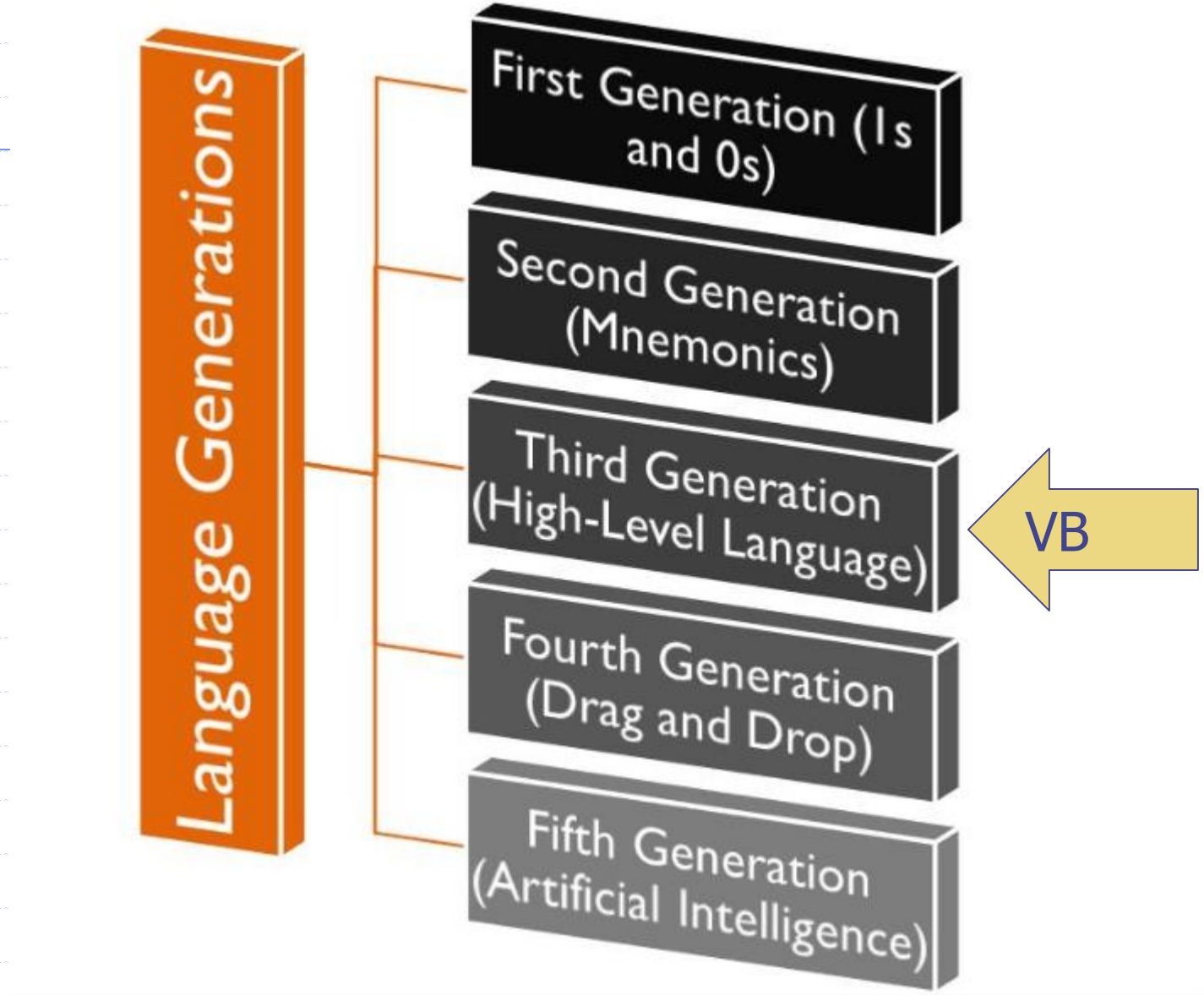
Increasingly user friendly



The Evolution of Programming Languages

Note that we refer to these as *first*, *second* and *third* generation languages.




Language Generations
First Generation (1s and 0s)
Second Generation (Mnemonics)
Third Generation (High-Level Language)
Fourth Generation (Drag and Drop)
Fifth Generation (Artificial Intelligence)
VB

2 1 solution Medium Priority 951 Views Last Modified: 2008-02-26

Is Visual basic a 3GL or 4GL??

Secondly, where can I get decent info on what defines these terms and their usages, etc.

Comment Share

Question by: Shaley2

2 Comments

Participants: 2

Accepted Solution by: vettranger

ID: 1510309 • 1999-05-11

LVL 8

Give that a definite 3GL.

4GLs are supposed to get into the neighborhood of natural language processing. Outside of its GUI features, VB is definitely a 'traditional' high level programming language, altho its detractors would question the use of high level with VB. ;-) (I'm not one of them BTW.)

2 1 solution Medium Priority 951 Views Last Modified: 2008-02-26

Is Visual basic a 3GL or 4GL??

Secondly, where can I get decent info on what defines these terms and their usages, etc.

Comment Share

Question by: Shaley2

Accepted Solution by: seahpc

ID: 1463034 • 1998-06-10

LVL 2

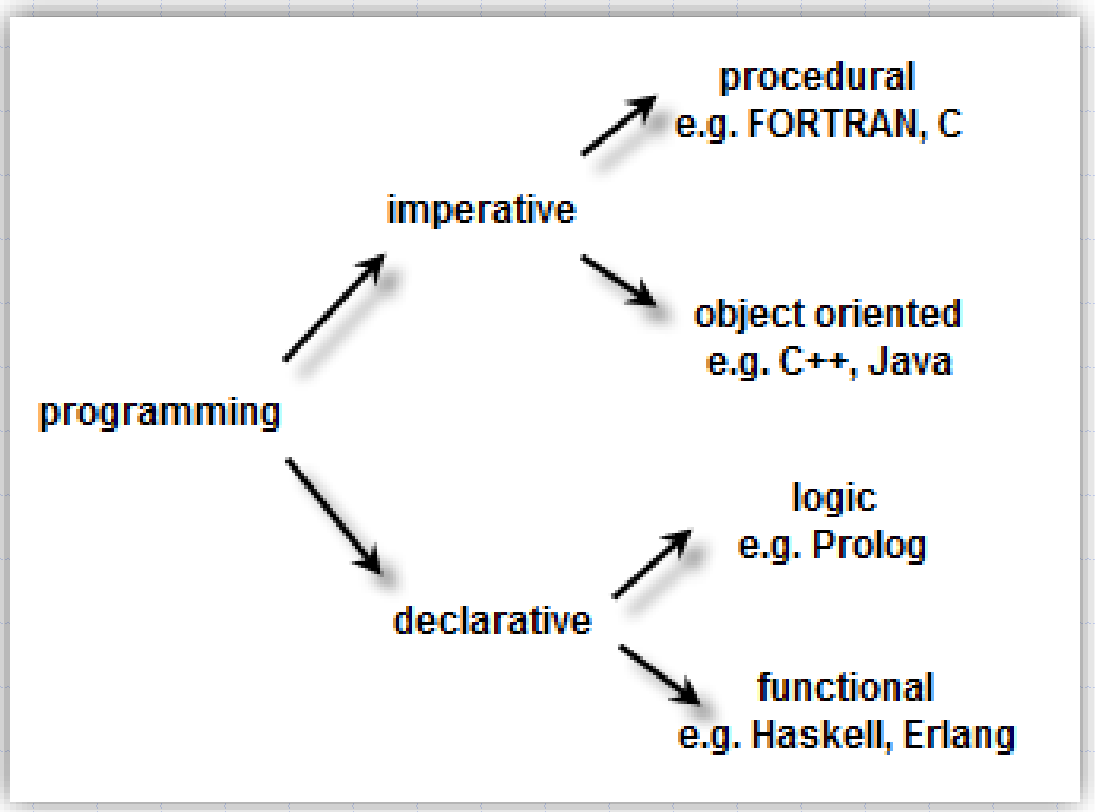
Yes VB is considered a 4GL .

It satisfy the feature of a 4GL . It allows the design of screens ,connection to Database and it is easy and fast to develop .

Some of 4GL Program that can be found are VB (Basic), Visual Cafe(Java), FoxPro .....

I think you are able to download their trial version but not the original . From the net . Go to their Website (Microsoft, Sun) and search for the product .

Classifying Programs by Implementation



Classifying Programs by Implementation

> ***Declarative programming*** *is a programming paradigm … that expresses the logic of a computation without describing its control flow.*
>
> ***Imperative programming*** *is a programming paradigm that uses statements that change a program's state.*

Classifying Programs by Implementation

> ***Declarative programming*** *is a programming paradigm … that expresses the logic of a computation without describing its control flow.*
>
> ***Imperative programming*** *is a programming paradigm that uses statements that change a program's state.*

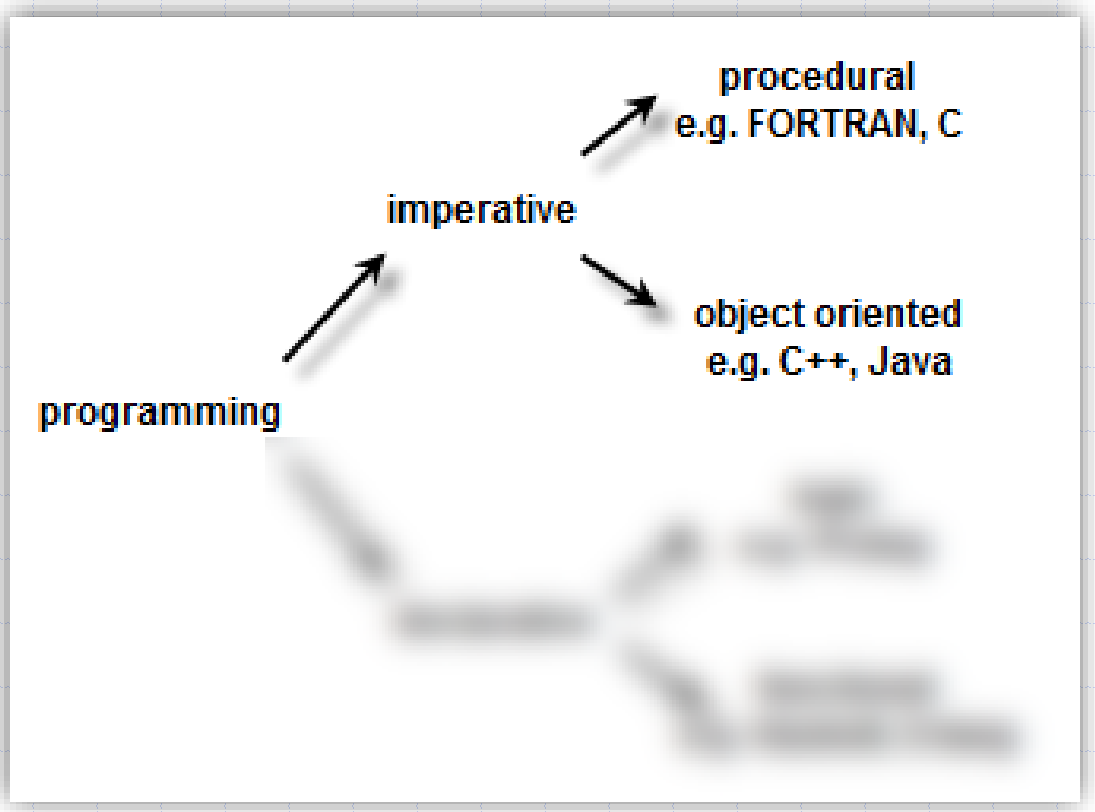
- *Declarative Programming* is like asking your friend to draw a landscape. *You don't care how they draw it,* that's up to them.
- *Imperative Programming* is like your friend listening to Bob Ross tell them how to paint a landscape. While good ole Bob Ross isn't exactly commanding, he is giving them *step by step directions* to get the desired result.

Classifying Programs by Implementation

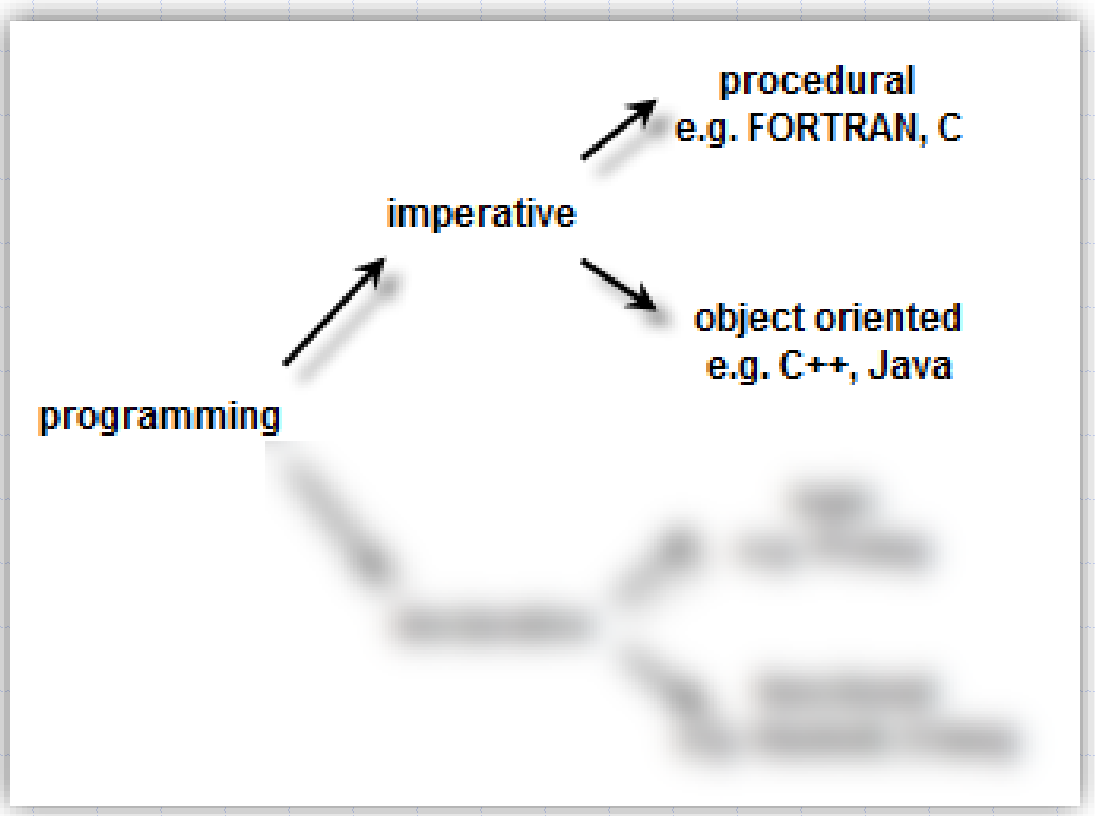
> ***Declarative programming*** *is a programming paradigm … that expresses the logic of a computation without describing its control flow.*
>
> ***Imperative programming*** *is a programming paradigm that uses statements that change a program's state.*

- *Declarative Programming* is like asking your friend to draw a landscape. *You don't care how they draw it,* that's up to them. **WHAT TO DO**
- *Imperative Programming* is like your friend listening to Bob Ross tell them how to paint a landscape. While good ole Bob Ross isn't exactly commanding, he is giving them *step by step directions* to get the desired result.**HOW TO DO IT**

Classifying Programs by Implementation



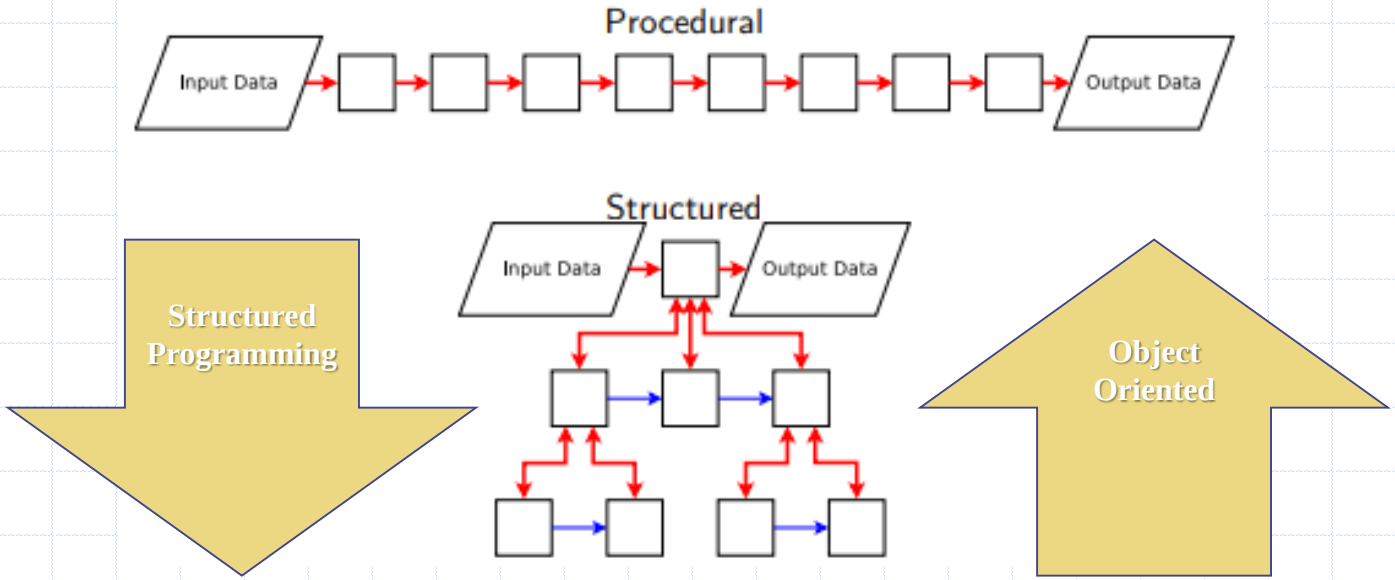
Classifying Programs by Implementation



Top-down

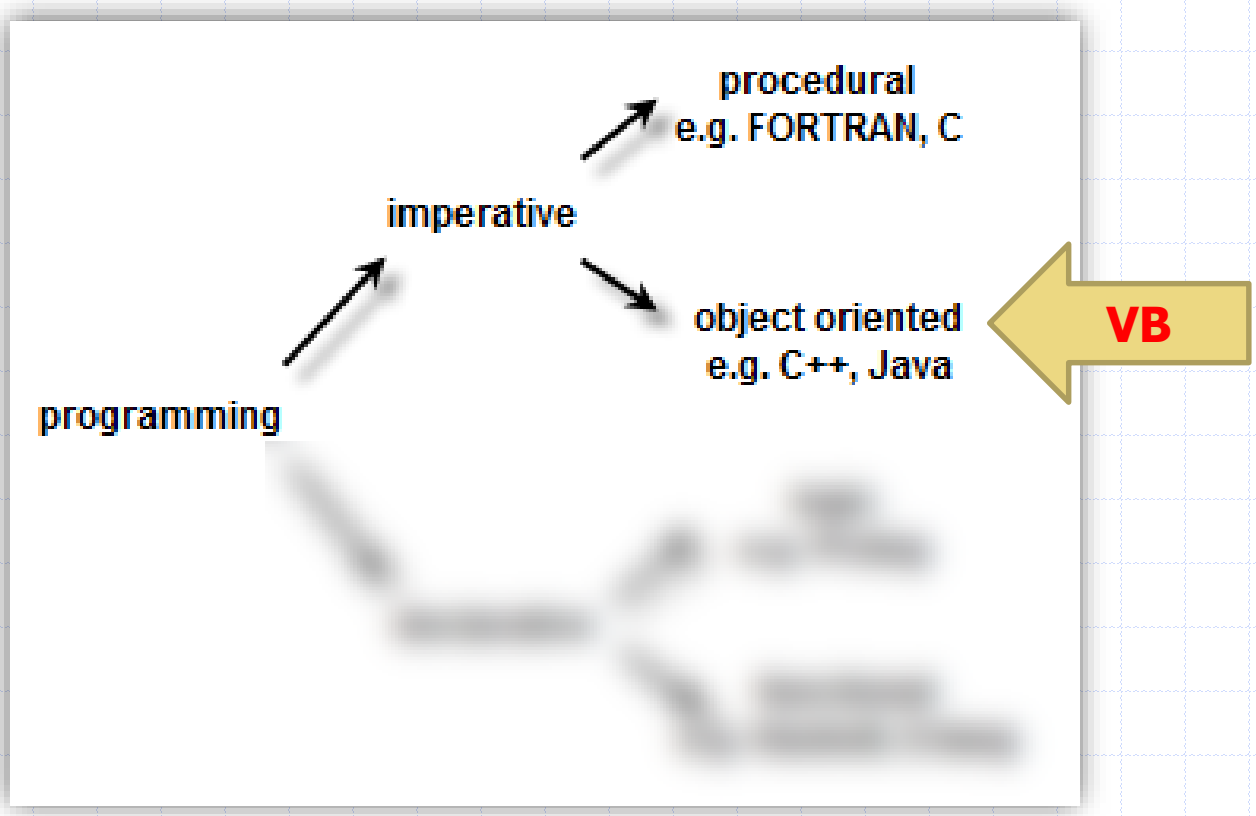
Bottom-up

OO Paradigm



https://wiki.scinet.utoronto.ca/wiki/images/4/49/Scientific-c%2B%2B.pdf

Classifying Visual Basic

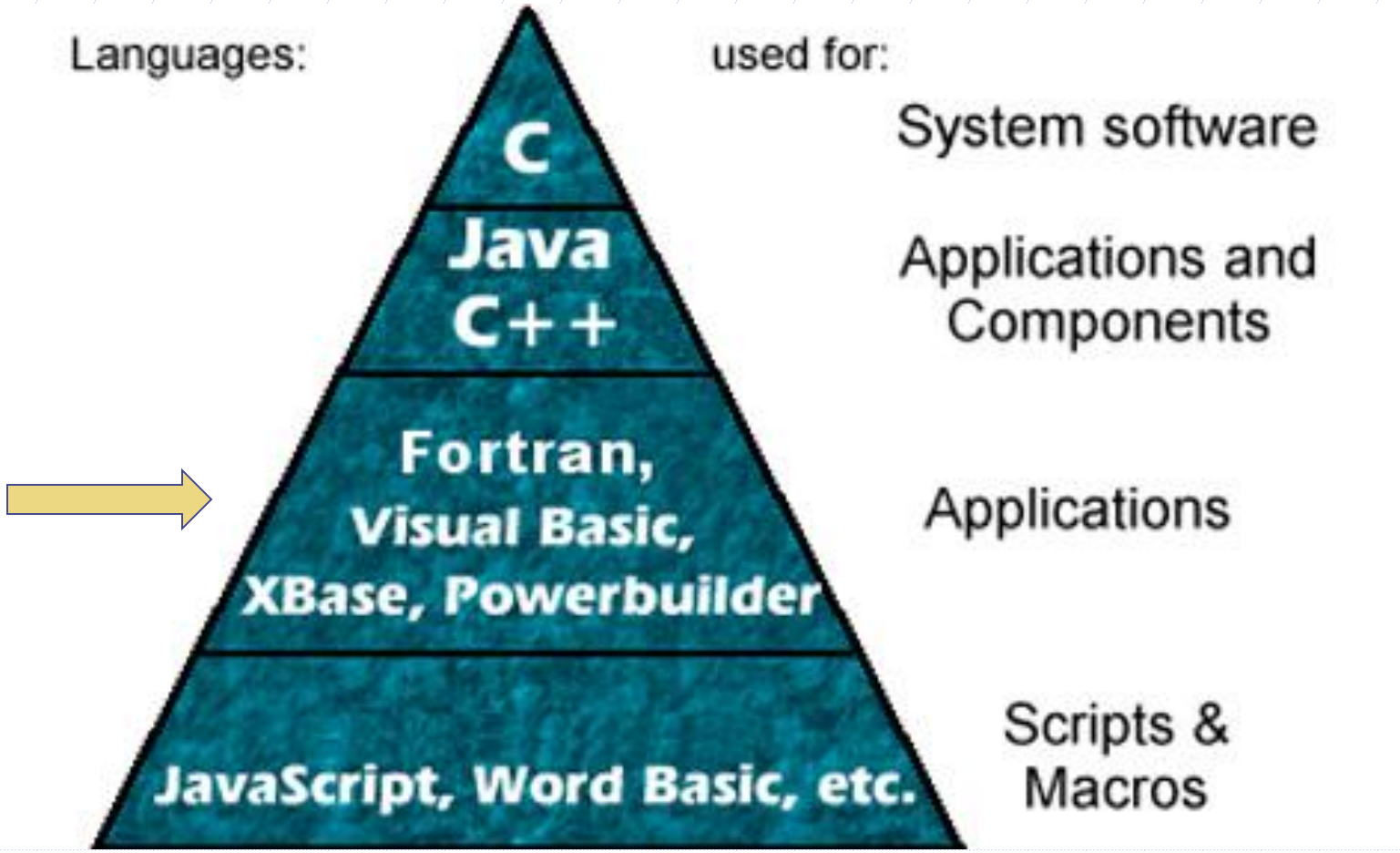


Another Classification Schema

Computer Languages are tailor-made for different problem classes. For example,

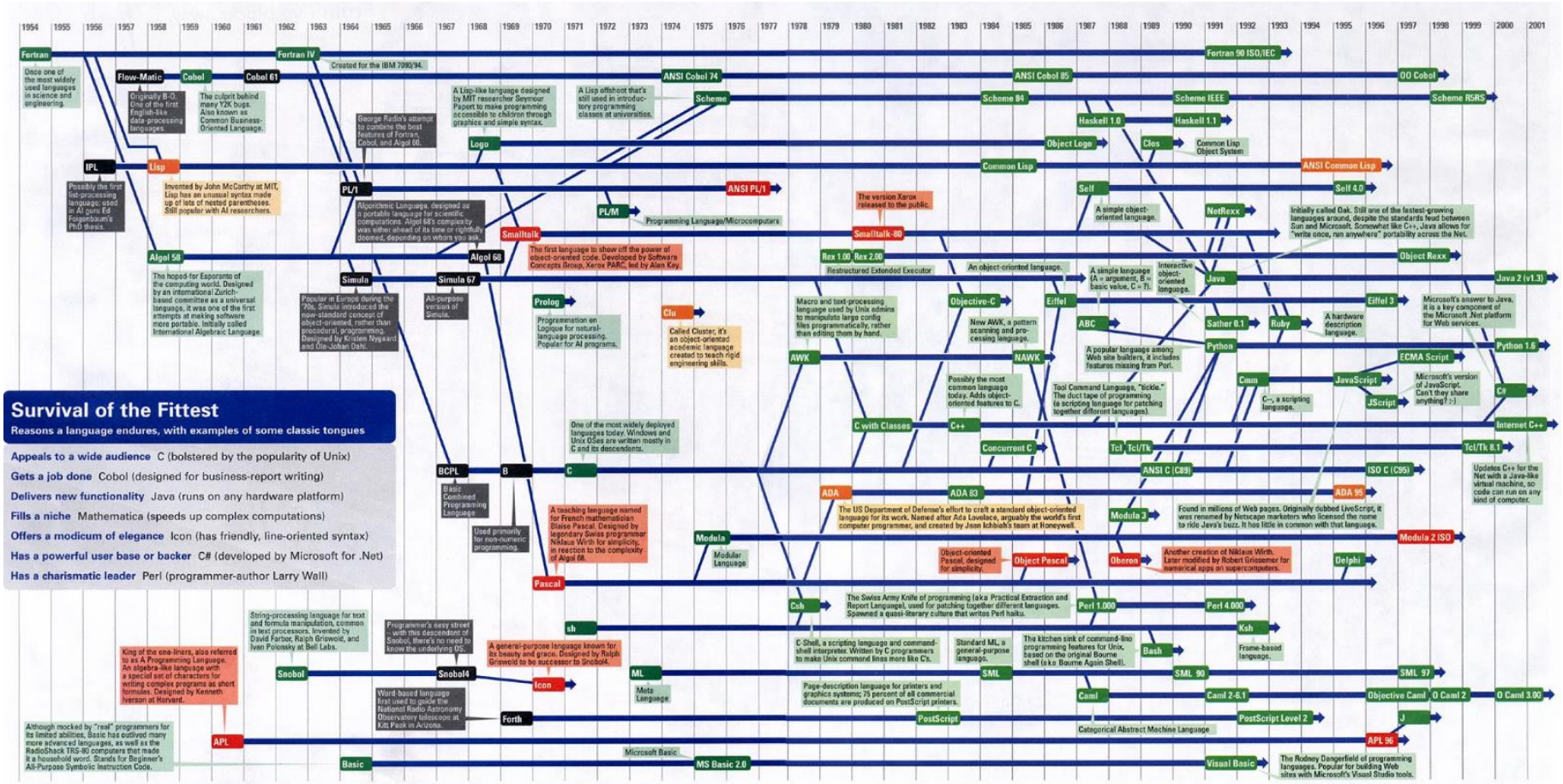
• FORTRAN	1954-57	numeric
• ALGOL 60	1958-60	numeric
• COBOL	1959-60	business
• APL	1956-60	vector/matrix math
• LISP	1956-62	symbols
• SNOBOL4	1962-66	strings
• PL/1	1963-64	general
• BASIC	1964	educational
• PASCAL	1971	educational

Classification by Problem Class

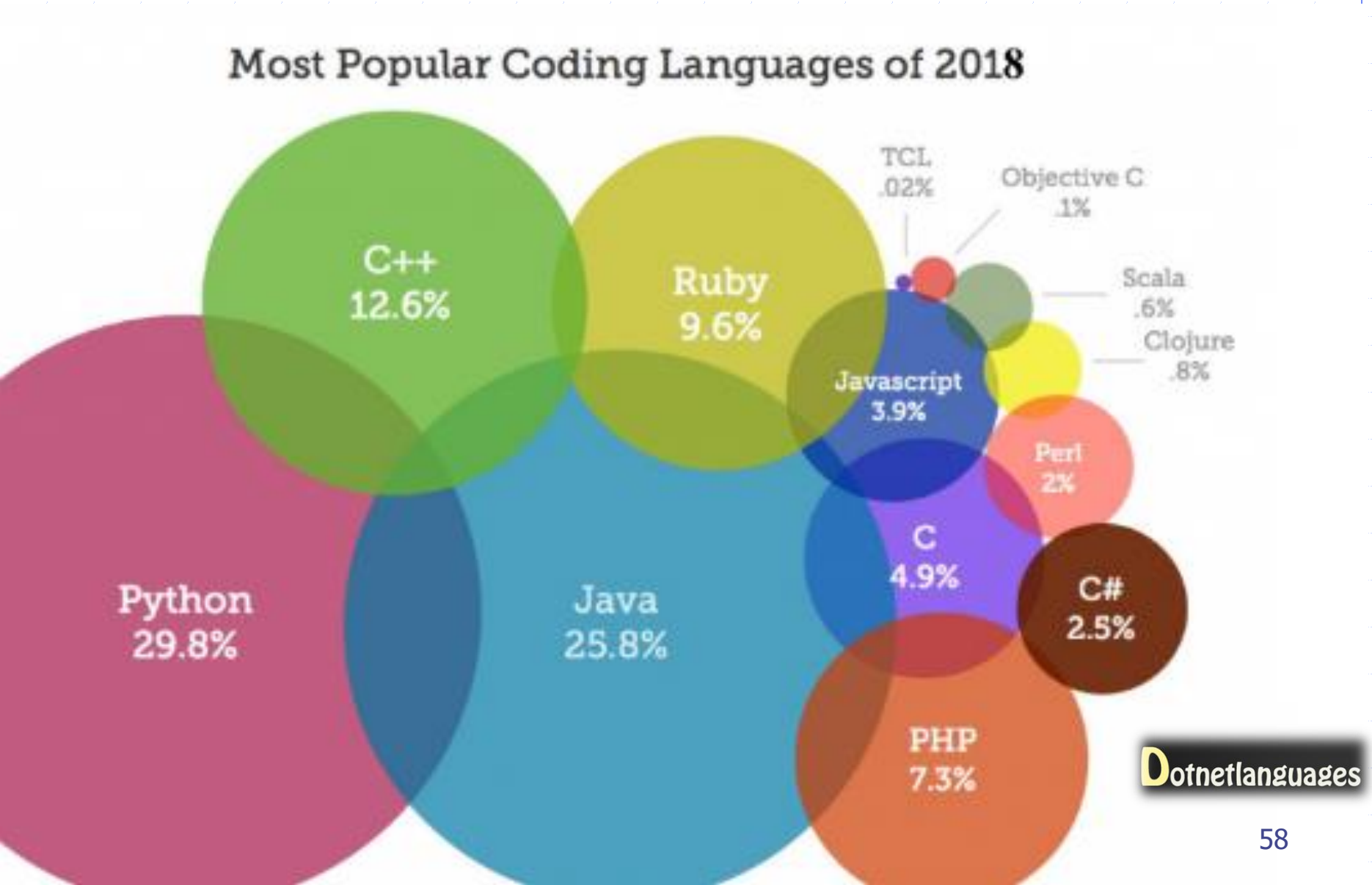


https://www.kullabs.com/classes/subjects/units/lessons/notes/note-detail/698

The Programming Language Zoo



http://www.digibarn.com/collections/posters/tongues/tongues.jpg


Most Popular Coding Languages of 2018
C++
12.6%
Ruby
9.6%
TCL
.02%
Objective C
.1%
Scala
.6%
Clojure
.8%
Javascript
3.9%
Perl
2%
C
4.9%
C#
2.5%
Python
29.8%
Java
25.8%
PHP
7.3%
Dotnetlanguages

Programming Interfaces and Environments

Programming Interfaces and Environments

- An **IDE** (*integrated development environment*) is a souped-up text editor with additional support for developing (such as forms designers, resource editors, *etc.*), compiling and debugging applications. *e.g.* Eclipse, **Visual Studio**.
- A **Library** is a chunk of code that you can call from your own code, to help you do things more quickly/easily. For example, a Bitmap Processing library will provide facilities for loading and manipulating bitmap images, saving you having to write all that code for yourself.
- An **API** (*application programming interface*) is a term meaning the functions/methods in a library that you can call to ask it to do things for you - the interface to the library. It is a set of routines, protocols, and tools for building software applications.
  An API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components.
- A **framework** is a big library that provides many services (rather than perhaps only one focused ability as most libraries do). For example, **.NET** provides an application framework - it provides most (if not all) of the services you need to write a vast range of applications - so one "library" provides support for pretty much everything you need to do. Often a framework supplies a base on which you build your own code, rather than you building an application that consumes library code.

API Versus IDE

- Both APIs and IDEs are used in the development of software.
- However, they have differences when considering what facilities they provide and how they behave.

What is the difference between API and IDE?

• APIs provide a communication layer between two applications; one being developed and one already developed.

• IDEs, being a development environment, are used to develop software programs from the scratch.

• APIs can be considered as software that provides a required service or as a library.

• IDEs comes with debugging, designing, version control and other useful tools to write programs.

• API is not a development environment.

Programming Tools

- An **SDK** (*software development kit*) is a library (often with extra tool applications, data files and sample code) that aid you in developing code that uses a particular system (e.g. extension code for using features of an operating system (Windows SDK), drawing 3D graphics via a particular system (DirectX SDK), writing add-ins to extend other applications (Office SDK), or writing code to make a device like an Arduino or a mobile phone do what you want)
- A **toolkit** is like an SDK - it's a group of tools (and often code libraries) that you can use to make it easier to access a device or system.

Front End Versus Back End

Front End Versus Back End

- In software engineering, the terms **front end** and **back end** refer to the separation of concerns between the presentation layer (*front end*), and the data access layer (*back end*) of a piece of software, or the physical infrastructure or hardware.
- In the client–server model, the client is usually considered the front end and the server is usually considered the back end, even when some presentation work is actually done on the server.

Front End Versus Back End

- The front end tool is the language used to create the user interface. The back end is the language used to create the functionality.
- For example, if you create a program that access a database, you may have VB as front end and SQL as back end (The user interface being in VB and the database functionality being SQL)

Examples of *Back-End* Languages

- Java (and other JVM languages like Scala, Groovy, Clojure)
- PHP
- .NET (C#, VB)
- Ruby
- Python
- Perl
- Javascript (Node JS)
- Actionscript (Flash Media Server)
- CoffeeScript
- C (CGI)
- Erlang
- and SQL for database queries

Examples of Browser-Based *Front-End* Languages

- HTML
- Javascript
- CSS
- Actionscript
- CoffeeScript (compiled to Javascript)
- XML-based languages (X3D, SMIL, SVG, DITA, some interpreted by the browser, others transformed using XSL)
- VBScript
- Silverlight
- Java (applets)

Examples of Native PC Desktop *Front-End* Languages

- Visual Basic 6 (from my experience with big enterprises, I bet a lot of those are still out there, just like Windows Vista)
- .NET
- Java (Swing apps)

Examples of Native Mobile Application *Front-End* Languages

- Objective C (for iOS apps)
- Java (Android apps)

Quora

Most Popular (Browser-Based) Web Application Combinations

- **Front-end**: HTML / Javascript / CSS
- **Back-end**: PHP / Java / Javascript / Ruby / .NET / Python

Quora

Summary: What is Visual Basic?

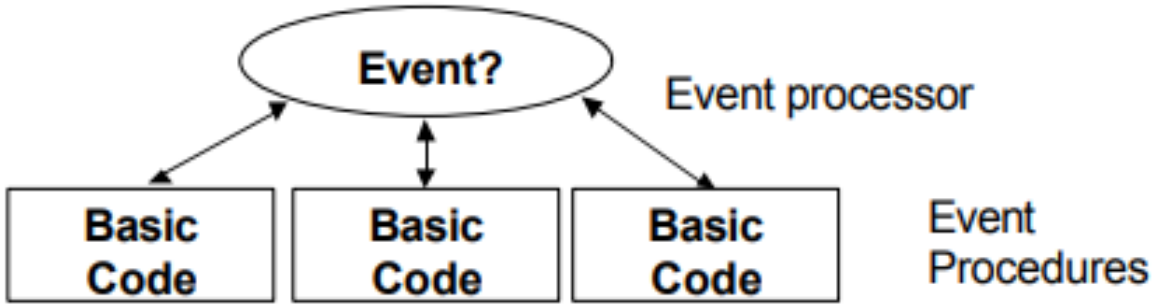
Summary: What is Visual Basic?

- **Visual Basic** is a tool that allows you to develop Windows (Graphic User Interface - **GUI**) applications. The applications have a familiar appearance to the user.

Summary: What is Visual Basic?

- **Visual Basic** is a tool that allows you to develop Windows (Graphic User Interface - **GUI**) applications. The applications have a familiar appearance to the user.

- Visual Basic is **event-driven**, meaning code remains idle until called upon to respond to some event (button pressing, menu selection, ...). Visual Basic is governed by an event processor. Nothing happens until an event is detected. Once an event is detected, the code corresponding to that event (event procedure) is executed. Program control is then returned to the event processor.



Summary: What is Visual Basic?

Features

Develop

Navigate, write, and fix your code fast

Debug

Debug, profile, and diagnose with ease

Test

Write high-quality code with comprehensive testing tools

Collaborate

Use version control, be agile, collaborate efficiently

Extend

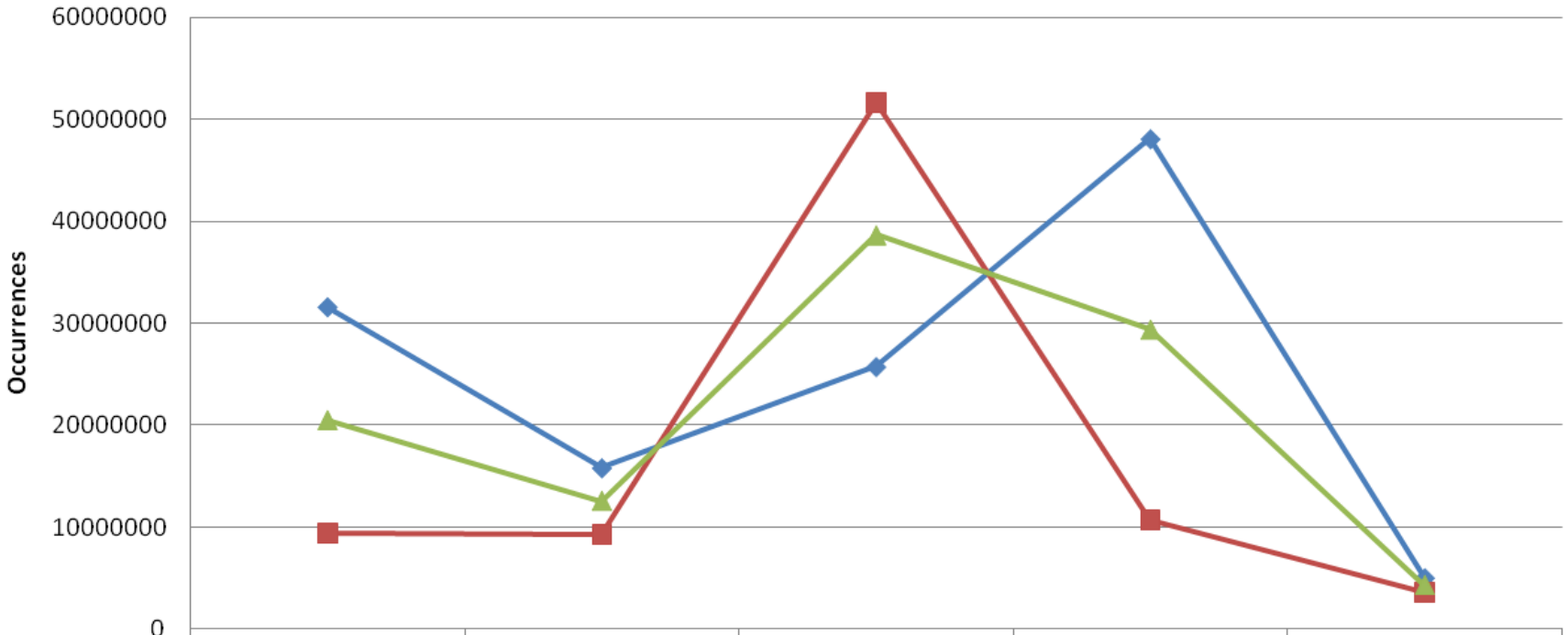
Choose from thousands of extensions to customize your IDE

https://visualstudio.microsoft.com/vs/features/

Summary: What is Visual Basic?

The main purpose of Visual Basic is for very user-friendly development of GUIs.

Best programming language in the world 2013-2014



	visual basic source code	VB source code	Visual Basic 6 plus VB6 source code	C++ source code	Java minus JavaScript source code
Google	31600000	15800000	25730000	48100000	5000000
Bing	9410000	9340000	51640000	10700000	3600000
Average	20505000	12570000	38685000	29400000	4300000

http://vb6awards.blogspot.ca/2014/01/best-programming-language-in-world-in.html

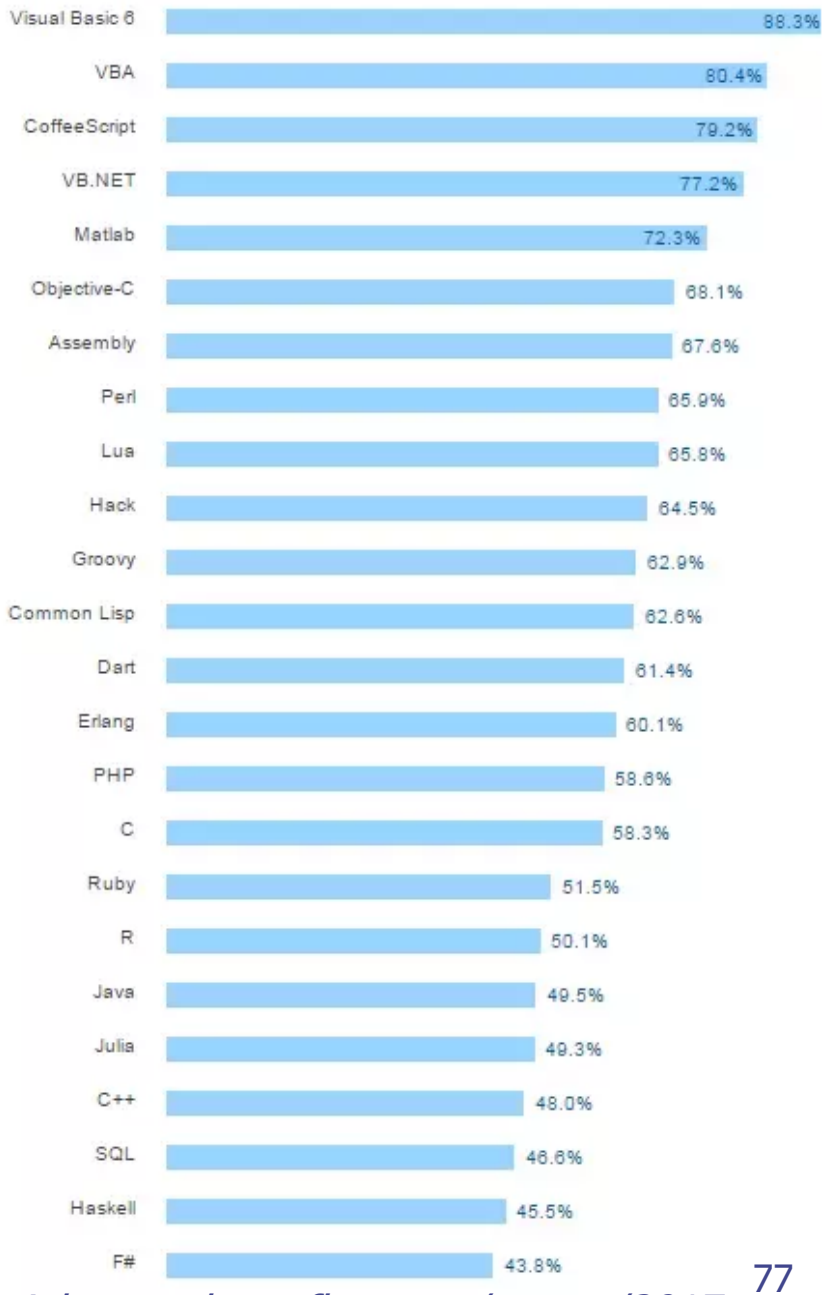
Most Dreaded

Mohammad Faizan Khan, Programmer at Private Firm
Answered Jun 16

Wait, **Who told you that VB is good programming language**.

OK, lets assume you are right but keep in mind it is **dead programming language** now. As recent Stackoverfow Survey (2017) stated

For the second year in a row, ***Visual Basic (for 2017, Visual Basic 6, specifically) ranked as the most dreaded language.*** *Most dreaded means that a high percentage of developers who are currently using the technology express no interest in continuing to do so.*

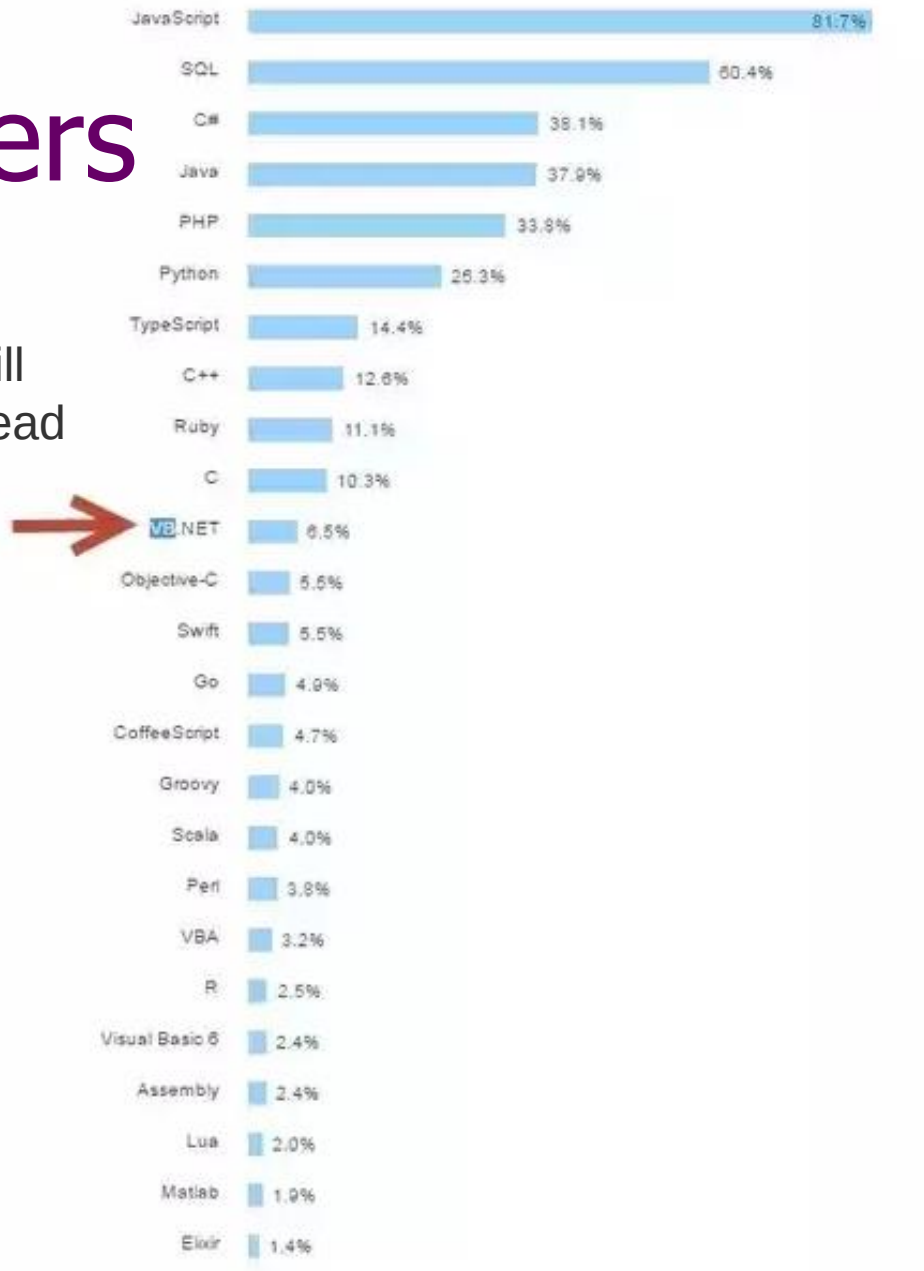


Quora

https://insights.stackoverflow.com/survey/2017

Web Developers

…if you are willing to learn any programming language then, I will recommend you not use VB instead JavaScript or C++ is best.”



Quora

https://insights.stackoverflow.com/survey/2017

10 Reasons Why Visual Basic is Better Than C#

> After having converted a whole lot of training materials based on VB.NET into C#, Andy 'Wise Owl' Brown decided to write a tongue-in-cheek rant whilst he could still remember the pain-points. 'Convert to VB.NET! You have nothing to lose but your semi-colons! '

Visual Basic is a better programming language than Visual C#. Who says so? This article! Here are 10 reasons why you should always choose VB over C#.

Note: C# is Microsoft's answer to Java (currently one of the most popular programming languages).

10 Reasons Why Visual Basic is Better Than C#

4 -Stupid symbols

C# was written by academics. It shows. Consider this table of C# symbols and their VB equivalents:

What you're trying to do	C# Symbol	VB Equivalent
Test if two conditions are both true	&&	and
Test if one or other condition is true	||	or
Test if a condition is not true	*!*	not
Concatenate two strings of text	+	&
Test if a condition is true within an if statement	==	=

Which column looks like it was designed by a real person?

Outstanding Resource

http://pages.cpsc.ucalgary.ca/~saul/vb_examples/#Tutorials

Excellent Website on Computer Languages

https://rosettacode.org/wiki/Rosetta_Code

Why is VB so popular? [closed]

28

☆

15

To me, Visual Basic seems clumsy, ugly, error-prone, and difficult to read. I'll let others explain why. While VB.net has clearly been a huge leap forward for the language in terms of features, I still don't understand why anyone would choose to code in VB over, say, C#.

However, I still see (what seems to be) the vast majority of commercial web apps from "MS shops" are built in VB. I could stand corrected on this, but VB still seems more popular than it deserves.

Can anyone help answer any (or all) of these questions:

- Am I missing something with VB? Is it easier to learn, or "friendlier" than C#? Are there features I don't know about?
- Why is VB/VB.net so frequently used today, especially in web projects?

Why is VB so popular? [closed]

28

☆

15

To me, Visual Basic seems clumsy, ugly, error-prone, and difficult to read. I'll let others explain why. While VB.net has clearly been a huge leap forward for the language in terms of features, I still don't understand why anyone would choose to code in VB over, say, C#.

However, I still see (what seems to be) the vast majority of commercial web apps from "MS shops" are built in VB. I could stand corrected on this, but VB still seems more popular than it deserves.

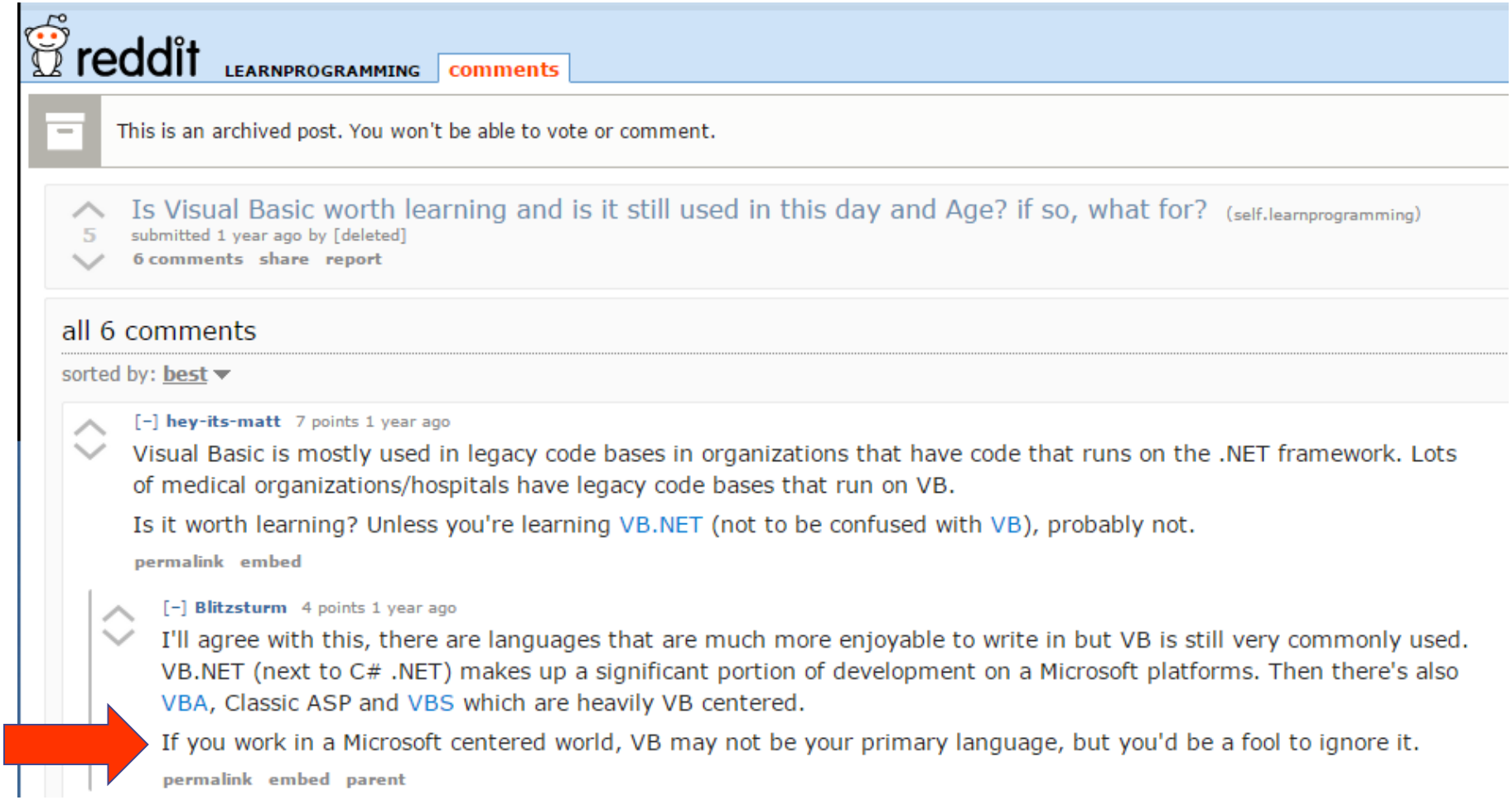
Can anyone help answer any (or all) of these questions:

- Am I missing something with VB? Is it easier to learn, or "friendlier" than C#? Are there features I don't know about?
- Why is VB/VB.net so frequently used today, especially in web projects?

42

I think that it depends on where you come from. When starting out as a programmer, I think that VB might be *easier* to read than C# for instance, since it relies more on words than symbols, which makes it easier to take in for regular people.

I was a VB programmer for many years and when .NET came I still worked in VB.NET for the first couple of years (didn't really see the point with C#). Now I have a few years of C# behind me and I sometimes find that VB.NET code takes a bit longer for me to "decode" than C# code does. Possibly because it relies more on words than symbols for some contructs...



https://www.reddit.com/r/learnprogramming/comments/4xxep2/is_visual_basic_worth_learning_and_is_it_still/

10 Reasons Why Visual Basic is Better Than C#

> After having converted a whole lot of training materials based on VB.NET into C#, Andy 'Wise Owl' Brown decided to write a tongue-in-cheek rant whilst he could still remember the pain-points. 'Convert to VB.NET! You have nothing to lose but your semi-colons! '

Visual Basic is a better programming language than Visual C#. Who says so? This article! Here are 10 reasons why you should always choose VB over C#.

1 – "Rose is a rose is a rose is a rose"

This is a quotation from Gertrude Stein's 1922 play *Geography and Plays*. However, the poetry wouldn't work in C#, because – unforgivably – it's a cASe-SeNSitIvE language. This is madness!

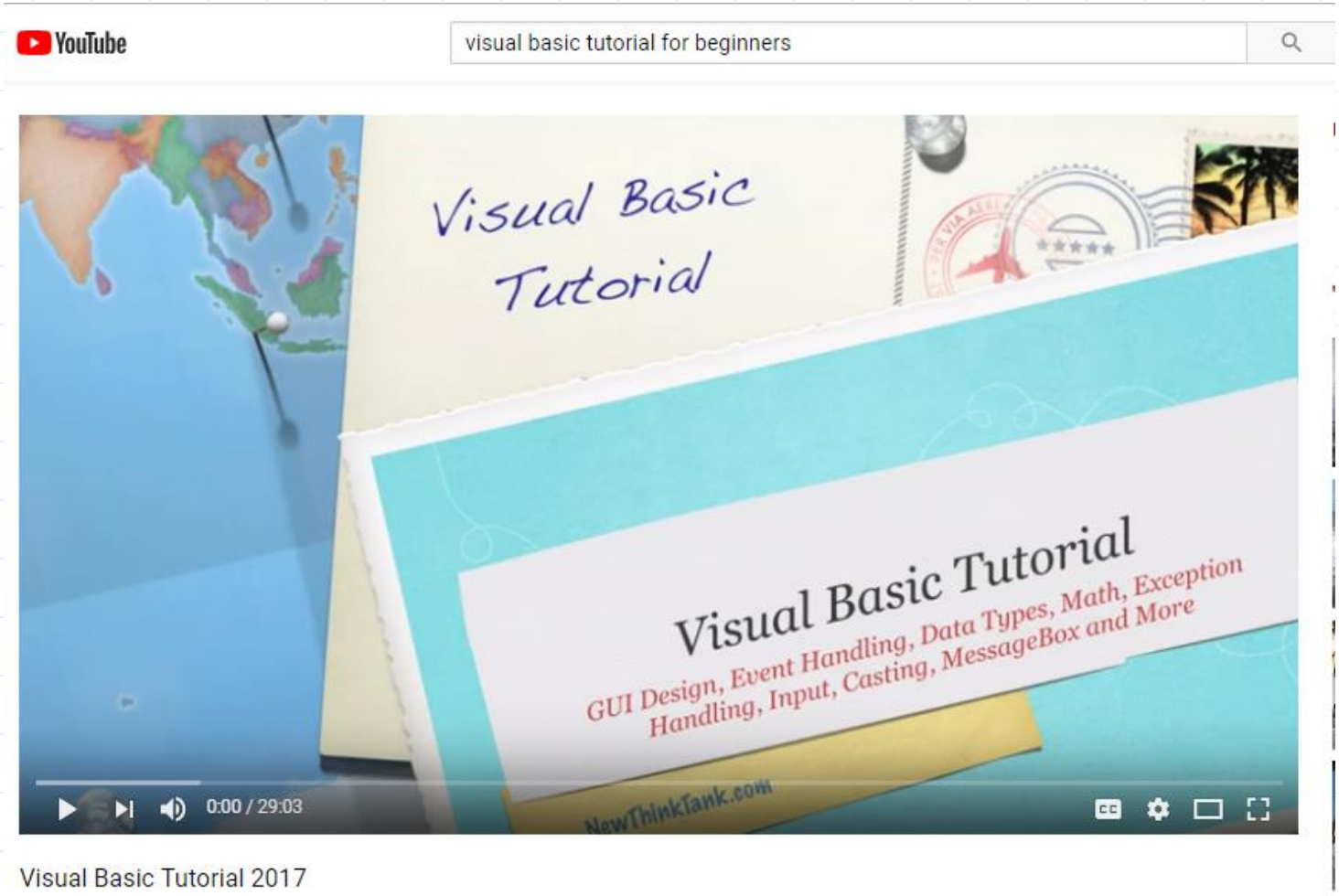
Before I start ranting, let me just acknowledge that case-sensitivity confers one (and only one) advantage – it makes it easier to name private and public properties:

Writing properties like this means that you can refer to the public Name property, and it's obvious what the private equivalent will be called (name).

```
// private version of variable
private string name = null;
public string Name
{
    get
    {
        return txtName.Text;
    }
    set
    {
        name = txtName.Text;
    }
}
```


https://www.red-gate.com/simple-talk/dotnet/net-framework/10-reasons-why-visual-basic-is-better-than-c/

Introductory Tutorials



https://www.youtube.com/watch?v=3FkWddODLno

Introduction Terminated