

# Introduction to Computer Programming

Instructor Özgür ZEYDAN  
BEU Dept. of Enve. Eng.  
<http://cevre.beun.edu.tr/zeydan/>

CIV 112 – Computer Programming  
Lecture Notes (1)

## Computer Programming

- A **computer** is a programmable machine. This means it can execute a programmed list of instructions and respond to new instructions that it is given.
- **Computer Programming** is the process of developing and implementing various sets of instructions to enable a computer to do a certain task.
- **Programs** are written to solve problems or perform tasks on a computer.

# Computer Programming

- **Programmers** translate the solutions or tasks into a language the computer can understand.
- As we write programs, we must keep in mind that the computer will only do what we instruct it to do.
- Because of this, we must be very careful and thorough with our instructions.



Instructor Özgür ZEYDAN

3

## First Computer Programmer: Ada Lovelace



**Ada Lovelace** is the first person to develop an **algorithm** for a machine.

- In 1842-1843, Lovelace translated an article about Charles Babbage's proposed Analytic Engine. In her notes, she describes an algorithm that is cited as **the first computer program**, making her the first computer programmer.
- She also theorized that the computer could, one day, **play music and chess**.
- **Ada**, a U.S. Department of Defense computer language, is **named in her honor**.

Instructor Özgür ZEYDAN

4

# Algorithm

- An **algorithm** is a list of instructions, procedures, or formulas used to solve a problem.
- The word derives from the name of the mathematician, Mohammed ibn-Musa al-Khwarizmi (**El-Harezmi**), (780 – 850).



# Pseudocode

- **Pseudocode** is a computer programming language that resembles plain English that cannot be compiled or executed, but explains a resolution to a problem.



## Source Code

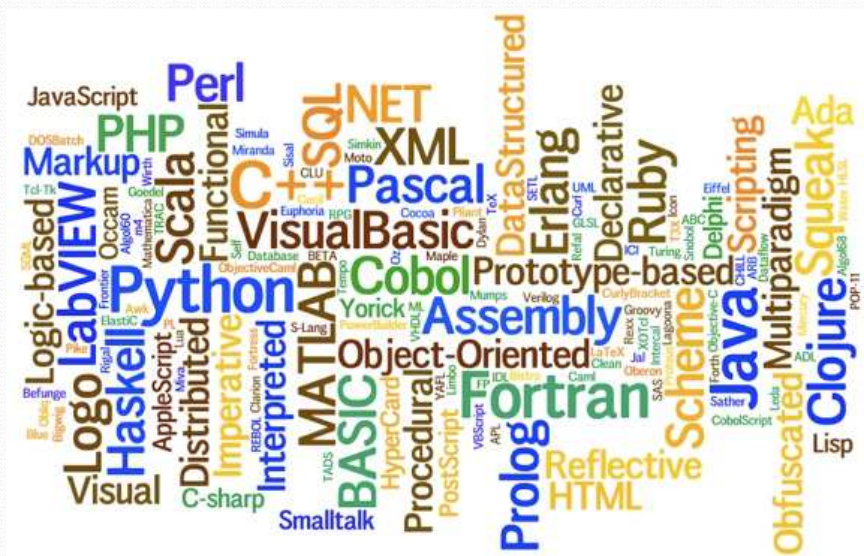
- The **source code** consists of the programming statements that are created by a programmer with a text editor or a visual programming tool and then saved in a file.
- For example, a programmer using the C language types in a desired sequence of C language statements using a text editor and then saves them as a named file.
- This file is said to contain the **source code**.

## Flowchart

- A **flowchart** is a formalized graphic representation of a logic sequence, work or manufacturing process, organization chart, or similar formalized structure.
- The purpose of a flow chart is to provide people with a common language or reference point when dealing with a project or process.
- **Flowcharts use simple geometric symbols and arrows to define relationships.**

# Programming Languages

- Computer programming is almost always done by means of Programming Language.
- There exists more than 2500 programming languages in the world.
- Some of them are known by only their developers!
- For further information:  
[http://en.wikipedia.org/wiki/List\\_of\\_programming\\_languages](http://en.wikipedia.org/wiki/List_of_programming_languages)



# History of Programming Languages



Instructor Özgür ZEYDAN

11

## 1st Generation Vacuum Tubes, Magnetic drums

### 2nd Generation Transistors

### 3rd Generation Integrated Circuit

### 4th Generation Chip

### 5th Generation

### 6th Generation

### 7th Generation

### 8th Generation

### 9th Generation

### 10th Generation

### 11th Generation

### 12th Generation

### 13th Generation

### 14th Generation

### 15th Generation

### 16th Generation

### 17th Generation

### 18th Generation

### 19th Generation

### 20th Generation

### 21st Generation

### 22nd Generation

### 23rd Generation

### 24th Generation

### 25th Generation

### 26th Generation

### 27th Generation

### 28th Generation

### 29th Generation

### 30th Generation

### 31st Generation

### 32nd Generation

### 33rd Generation

### 34th Generation

### 35th Generation

### 36th Generation

### 37th Generation

### 38th Generation

### 39th Generation

### 40th Generation

### 41st Generation

### 42nd Generation

### 43rd Generation

### 44th Generation

### 45th Generation

### 46th Generation

### 47th Generation

### 48th Generation

### 49th Generation

### 50th Generation

### 51st Generation

### 52nd Generation

### 53rd Generation

### 54th Generation

### 55th Generation

### 56th Generation

### 57th Generation

### 58th Generation

### 59th Generation

### 60th Generation

### 61st Generation

### 62nd Generation

### 63rd Generation

### 64th Generation

### 65th Generation

### 66th Generation

### 67th Generation

### 68th Generation

### 69th Generation

### 70th Generation

### 71st Generation

### 72nd Generation

### 73rd Generation

### 74th Generation

### 75th Generation

### 76th Generation

### 77th Generation

### 78th Generation

### 79th Generation

### 80th Generation

### 81st Generation

### 82nd Generation

### 83rd Generation

### 84th Generation

### 85th Generation

### 86th Generation

### 87th Generation

### 88th Generation

### 89th Generation

### 90th Generation

### 91st Generation

### 92nd Generation

### 93rd Generation

### 94th Generation

### 95th Generation

### 96th Generation

### 97th Generation

### 98th Generation

### 99th Generation

### 100th Generation

### 101st Generation

### 102nd Generation

### 103rd Generation

### 104th Generation

### 105th Generation

### 106th Generation

### 107th Generation

### 108th Generation

### 109th Generation

### 110th Generation

### 111th Generation

### 112th Generation

### 113th Generation

### 114th Generation

### 115th Generation

### 116th Generation

### 117th Generation

### 118th Generation

### 119th Generation

### 120th Generation

### 121st Generation

### 122nd Generation

### 123rd Generation

### 124th Generation

### 125th Generation

### 126th Generation

### 127th Generation

### 128th Generation

### 129th Generation

### 130th Generation

### 131st Generation

### 132nd Generation

### 133rd Generation

### 134th Generation

### 135th Generation

### 136th Generation

### 137th Generation

### 138th Generation

### 139th Generation

### 140th Generation

### 141st Generation

### 142nd Generation

### 143rd Generation

### 144th Generation

### 145th Generation

### 146th Generation

### 147th Generation

### 148th Generation

### 149th Generation

### 150th Generation

### 151st Generation

### 152nd Generation

### 153rd Generation

### 154th Generation

### 155th Generation

### 156th Generation

### 157th Generation

### 158th Generation

### 159th Generation

### 160th Generation

### 161st Generation

### 162nd Generation

### 163rd Generation

### 164th Generation

### 165th Generation

### 166th Generation

### 167th Generation

### 168th Generation

### 169th Generation

### 170th Generation

### 171st Generation

### 172nd Generation

### 173rd Generation

### 174th Generation

### 175th Generation

### 176th Generation

### 177th Generation

### 178th Generation

### 179th Generation

### 180th Generation

### 181st Generation

### 182nd Generation

### 183rd Generation

### 184th Generation

### 185th Generation

### 186th Generation

### 187th Generation

### 188th Generation

### 189th Generation

### 190th Generation

### 191st Generation

### 192nd Generation

### 193rd Generation

### 194th Generation

### 195th Generation

### 196th Generation

### 197th Generation

### 198th Generation

### 199th Generation

### 200th Generation

### 201st Generation

### 202nd Generation

### 203rd Generation

### 204th Generation

### 205th Generation

### 206th Generation

### 207th Generation

### 208th Generation

### 209th Generation

### 210th Generation

### 211th Generation

### 212th Generation

### 213th Generation

### 214th Generation

### 215th Generation

### 216th Generation

### 217th Generation

### 218th Generation

### 219th Generation

### 220th Generation

### 221st Generation

### 222nd Generation

### 223rd Generation

### 224th Generation

### 225th Generation

### 226th Generation

### 227th Generation

### 228th Generation

### 229th Generation

### 230th Generation

### 231st Generation

### 232nd Generation

### 233rd Generation

### 234th Generation

### 235th Generation

### 236th Generation

### 237th Generation

### 238th Generation

### 239th Generation

### 240th Generation

### 241st Generation

### 242nd Generation

### 243rd Generation

### 244th Generation

### 245th Generation

### 246th Generation

### 247th Generation

### 248th Generation

</



## Programming Language Generations

- **1GL or first-generation language** was (and still is) **machine language** or the level of instructions and data that the processor is actually given to work on.
- **2GL or second-generation language** is **assembler** (sometimes called "assembly") language.

## Programming Language Generations

- **3GL or third-generation language** is a "**high-level**" programming language, such as PL/I, C, or Java. A compiler converts the statements of a specific high-level programming language into machine language. A 3GL language requires a considerable amount of programming knowledge.
- **4GL or fourth-generation language** is designed to be closer to natural language than a 3GL language. Languages for accessing databases are often described as 4GLs.

## Programming Language Generations

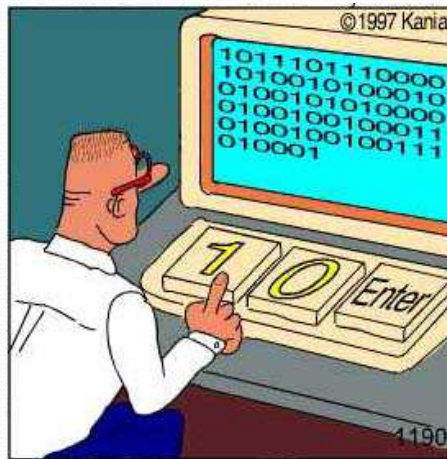
- **5GL or fifth-generation language** is programming that uses a **visual or graphical development interface** to create source language that is usually compiled with a 3GL or 4GL language compiler. Microsoft, Borland, IBM, and other companies make 5GL visual programming products for developing applications in Java, for example. Visual programming allows you to easily envision object-oriented programming class hierarchies and drag icons to assemble program components.

## Machine Code (machine language)

- **Machine code**, also known as machine language, is the elemental language of computers, comprising a long sequence of binary digital zeros and ones (bits).
- Sometimes referred to as machine code or object code, machine language is a collection of **binary digits or bits** that the computer reads and interprets. **Machine language is the only language a computer is capable of understanding.**



## Machine Code



Real programmers code in binary.

## Assembly Language

- Sometimes referred to as assembly or ASL, assembly language is a low-level programming language used to interface with computer hardware.
- Assembly language uses structured commands as substitutions for numbers allowing humans to read the code easier than looking at binary. Although easier to read than binary, assembly language is a difficult language and is usually substituted for a higher language such as C.

## Low-level Languages

- Low-level languages have the advantage that they can be written to take advantage of any peculiarities in the architecture of the central processing unit (CPU).
- Thus, a program written in a low-level language can be extremely efficient, making optimum use of both computer memory and processing time.
- However, to write a low-level program takes a substantial amount of time, as well as a clear understanding of the inner workings of the processor itself. Therefore, low-level programming is typically used only for very small programs, or for segments of code that are highly critical and must run as efficiently as possible.

[http://www.play-hookey.com/computers/language\\_levels.html](http://www.play-hookey.com/computers/language_levels.html)

Instructor Özgür ZEYDAN

19

## High-level Languages

- High-level languages permit faster development of large programs. The final program as executed by the computer is not as efficient, but the savings in programmer time generally far outweigh the inefficiencies of the finished product.
- This is because the cost of writing a program is nearly constant for each line of code, regardless of the language.
- Thus, a high-level language where each line of code translates to 1-10 machine instructions costs only one tenth as much in program development as a low-level language where each line of code represents only a single machine instruction.


[http://www.play-hookey.com/computers/language\\_levels.html](http://www.play-hookey.com/computers/language_levels.html)

Instructor Özgür ZEYDAN

20

# First High Level Language

**FIRST HIGH LEVEL COMPUTER PROGRAMMING LANGUAGE USED: FORTRAN**



- Invented by John Backus of IBM in 1954.
- It was released commercially in 1957.
- A high level programming language is one that is far removed from the computer's instruction architecture. It is typically more user friendly than low level programming languages.

Backus, 1954

Fortran

Fortran user manual

Design and Illustration  
By Ellie Koning

Instructor Özgür ZEYDAN

21

## Programming Language Popularities

Position Jan 2013	Position Jan 2012	Delta in Position	Programming Language	Ratings Jan 2013	Delta Jan 2012	Status
1	2	↑	C	17.855%	+0.89%	A
2	1	↓	Java	17.417%	-0.05%	A
3	5	↑↑	Objective-C	10.283%	+3.37%	A
4	4	=	C++	9.140%	+1.09%	A
5	3	↓↓	C#	6.196%	-2.57%	A
6	6	=	PHP	5.546%	-0.16%	A
7	7	=	(Visual) Basic	4.749%	+0.23%	A
8	8	=	Python	4.173%	+0.96%	A
9	9	=	Perl	2.264%	-0.50%	A
10	10	=	JavaScript	1.976%	-0.34%	A
11	12	↑	Ruby	1.775%	+0.34%	A
12	24	↑↑↑↑↑↑↑↑	Visual Basic .NET	1.043%	+0.56%	A
13	13	=	Lisp	0.953%	-0.16%	A
14	14	=	Pascal	0.932%	+0.14%	A
15	11	↓↓↓	Delphi/Object Pascal	0.919%	-0.65%	A
16	17	↑	Ada	0.651%	+0.02%	B
17	23	↑↑↑↑↑	MATLAB	0.641%	+0.13%	B
18	20	↑↑	Lua	0.633%	+0.07%	B
19	21	↑↑	Assembly	0.629%	+0.08%	B
20	72	↑↑↑↑↑↑↑↑	Bash	0.613%	+0.49%	B













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

Instructor Özgür ZEYDAN

22



## Comparison of Programming Languages

 C++	 JavaScript
 Java/C#	 PHP (Without MySQL)
 Ruby	 Pascal
 Perl	 Lisp
 Visual Basic	 Haskell
 Python	 C

<http://anggriawan.web.id/2011/06/if-programming-languages-were.html>

Instructor Özgür ZEYDAN

23

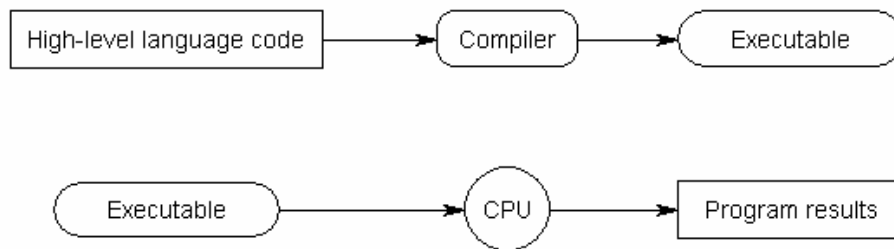
## Compiler

- A **compiler** is a special program that processes statements written in a particular programming language and turns them into machine language or "code" that a computer's processor uses.
- After you write a program, your source language statements are compiled into machine code that is stored as an executable file.
- Scripting languages like Perl and PHP do not need to be compiled.

Instructor Özgür ZEYDAN

24

# Compiler

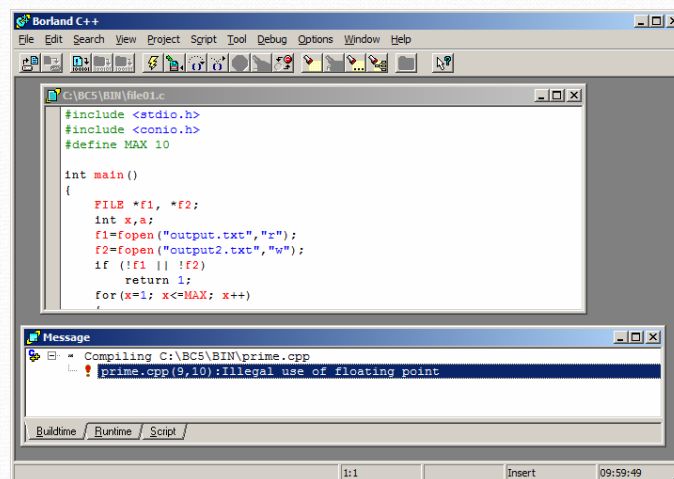


<http://www.learncpp.com/cpp-tutorial/02-introduction-to-programming-languages/>

Instructor Özgür ZEYDAN

25

# Borland C++ Compiler



Compiler is available from following page after free registration:  
<http://forms.embarcadero.com/forms/BCC32CompilerDownload>

Instructor Özgür ZEYDAN

26

## Structured Programming (modular programming)

- **Structured programming** (sometimes known as modular programming) is a subset of procedural programming that enforces a logical structure on the program being written to make it more efficient and easier to understand and modify. Certain languages such as Ada, Pascal, and dBASE are designed with features that encourage or enforce a logical program structure.
- Structured programming frequently employs a top-down design model, in which developers map out the overall program structure into separate subsections.
- Program flow follows a simple hierarchical model that employs looping constructs such as "for", "repeat", and "while". Use of the "Go To" statement is discouraged.
- Structured programming was first suggested by Corrado Bohm and Guiseppe Jacopini. The two mathematicians demonstrated that any computer program can be written with just three structures: **decisions**, **sequences**, and **loops**.

## Object-Oriented Programming (OOP)

- **Object-oriented programming** (OOP) is a programming language model organized around "objects" rather than "actions" and **data rather than logic**.
- Historically, a program has been viewed as a logical procedure that takes input data, processes it, and produces output data.



## Classification of Programming Languages

- **Procedure-oriented programming**
  - COBOL, FORTRAN, Pascal and C
- **Object oriented programming**
  - Objective C, C++, Java, and PHP

## Integrated Development Environment (IDE)

An IDE or **Integrated Development Environment** is a software program that is designed to help programmers and developers build software.

Most IDEs include:

- a source code editor
- a compiler
- build automation tools
- a debugger

# Debugger



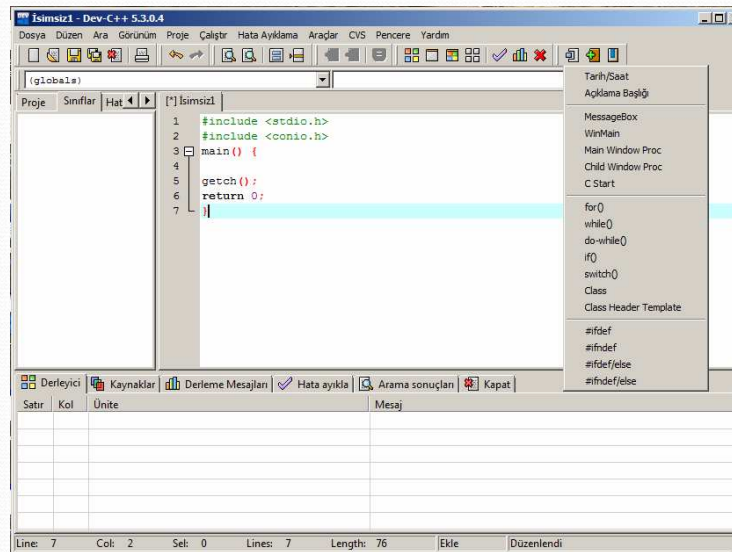
- A special program used to find errors (bugs) in other programs. A debugger allows a programmer to stop a program at any point and examine and change the values of variables.
- <http://www.webopedia.com/TERM/D/debugger.html>

# Graphical User Interface (GUI)

- A GUI is a graphical (rather than purely textual) user interface to a computer.
- Elements of a GUI include textboxes, buttons, pulldown menus, list and combo boxes



# DEV C++ IDE

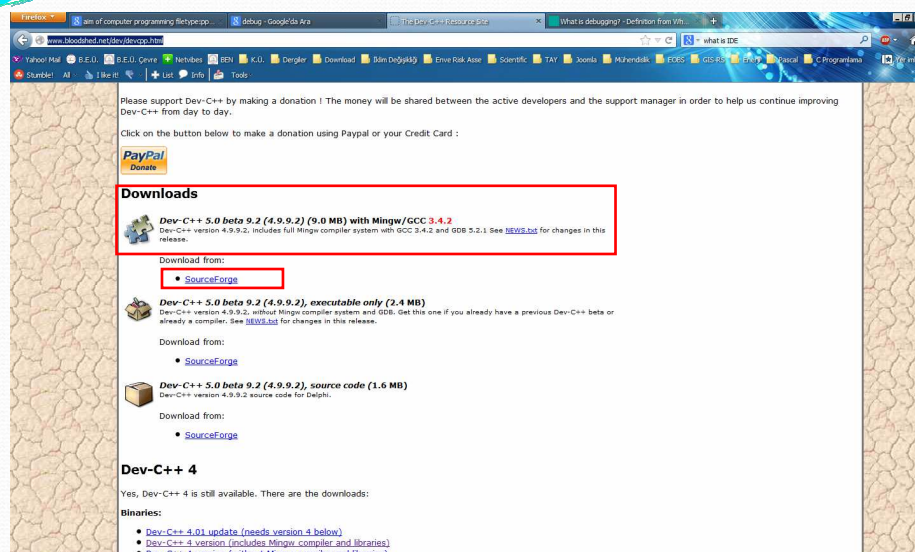


<http://www.bloodshed.net/dev/devcpp.html>

Instructor Özgür ZEYDAN

33

# DEV C++ IDE download page



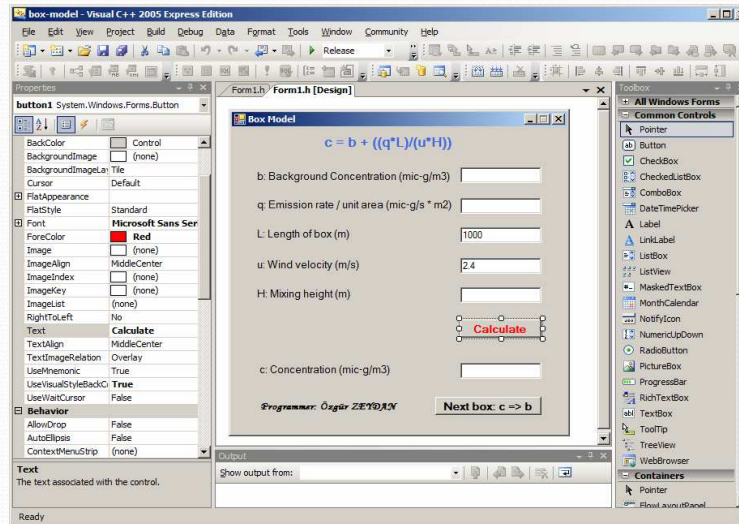
<http://www.bloodshed.net/dev/devcpp.html>

Instructor Özgür ZEYDAN

34



# Visual C++ Express Edition



<http://www.microsoft.com/visualstudio/tur/downloads#d-2010-express>

Instructor Özgür ZEYDAN

35

## Software Development Languages

- C
- C++ (C-plus-plus)
- C# (C-Sharp)
- Pascal
- Delphi
- Visual Basic

Instructor Özgür ZEYDAN

36

# Web Languages

- HTML (Hyper Text Markup Language)
- XML (Extensible Markup Language)
- Javascript
- VBScript
- PHP (Hypertext Preprocessor)
- Java
- ASP (Active Server Pages)

<http://landofcode.com/programming-intro/computer-programming-languages.php>

Instructor Özgür ZEYDAN

37

# Important People in Computer Programming



**Charles Babbage**

{He first came up with the idea of difference engine & analytical engine and is regarded as father of computer}

**John Backus**

{He is well known for the development of FORTRAN and ALGOL. He is also the inventor of Backus-Naur form and has also helped to popularize functional level programming}



**Alan Turing**

{He is well known for the Halting problem, Turing machines, cryptanalysis of Enigma & Turing test. Turing award is given annually for exceptional work in the field of computing}

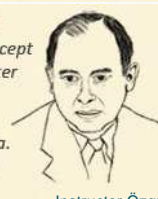


**Dennis Ritchie**

{He is the creator of C programming language and was also amongst the key developers of UNIX operating system. He received the Turing award in 1983}

**John von Neumann**

{He came up with the concept of stored program computer that uses a CPU and a separate storage to hold both instructions and data. This is also known as von Neumann architecture}



**Ken Thompson**

{He is well known as the principal creator of the UNIX operating system and is also the co-creator of the Go programming language}

Instructor Özgür ZEYDAN

38

## Important People in Computer Programming

### Bjarne Stroustrup

{He is well known for the creation and development of C++ programming language and currently holds the college of engineering chair in computer science at Texas A&M.}



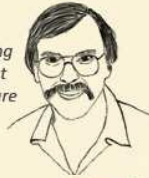
### Guido van Rossum

{He is well known as the author of Python programming language and is currently employed by Google.}



### Larry Wall

{He is well known for the creation of Perl programming language and is also the first recipient of the Free Software Foundation Award for the Advancement of Free Software}



### Richard Stallman

{He is the creator of Emacs editor and the lead architect and organizer of the GNU project. He has been actively involved in the free software movement}



### Linus Torvalds

{He is best known for having initiated the development of Linux Kernel and the Git revision control system. He is also a strong supporter of Open Source software}



Instructor Özgür ZEYDAN

 **GeekGraphic**  
part of what - a - geek.com  
Designed By **Adit Gupta**

39

## List of other reference web pages:

- <http://www.techterms.com/>
- <http://www.computerhope.com>
- <http://whatis.com/>

Instructor Özgür ZEYDAN

40