



Overview of Computer Software & Programming Language

Er. Shiva K. Shrestha (HoD)

Department of Computer Engineering,

Khwopa College of Engineering



C Evaluation

- ▶ Internal Marks (20)
 - ▶ First Assessment (5)
 - ▶ Assignment (4)
 - ▶ Class Test (1)
 - ▶ Attendance (2)
 - ▶ Final Assessment (8)
- ▶ Practical Marks (30)
 - ▶ Lab Test (5)
 - ▶ Lab Attendance (4)
 - ▶ Lab Report (9)
 - ▶ Viva (2)
 - ▶ Final Lab Exam (10)
- ▶ Mini Project (20)
 - ▶ Proposal (8)
 - ▶ Hard-copy Proposal (2)
 - ▶ Presentation (2)
 - ▶ Team-work (2)
 - ▶ Viva (2)
 - ▶ Final Defense (12)
 - ▶ Hard-copy Proposal (3)
 - ▶ Presentation (3)
 - ▶ Team-work (3)
 - ▶ Viva (3)

GitHub.com/KCE/C – Signup & Watch

3


Er. Shiva K. Shrestha (HoD, Computer De
10/3/2018


https://github.com/join?source=header-home

Features Business Explore Marketplace Pri

Join GitHub

The best way to design, build, and ship software.

 **Step 1:**
Create personal account

 **Step 2:**
Choose your profile picture

Create your personal account

Username

This will be your username. You can add the name of your organization later.

Email address

We'll occasionally send updates about your account to this inbox. We'll never share your email address with anyone.

Password

Make sure it's at least 7 characters, including a number, and a lowercase letter.

Verify account

https://github.com/KCE/C

Pull requests Issues Marketplace Explore

KCE / C

Unwatch 1 Star 1 Fork 2

Code Issues 5 Pull requests 0 Projects 0 Wiki Insights

C-Programming [Learning Material]

7 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

ErSKS New Updates Latest commit a5681fe 17 minutes ago

C_Labsheets_1_9.pdf	C Programming Labsheets	2 hours ago
C_Programming_Syllabus.pdf	Syllabus - C Programming	
Labsheet_1_v2.pdf	Labsheet#1	
README.md	New Updates	

README.md

C-Programming [Learning Materials]

Contact Email: c.khwopa@gmail.com or computer.khwopa@gmail.com

Computer Programming Notes

- Chapter 1: Overview of Computer Software & Programming Languages
- Chapter 1: Overview of Computer Software & Programming Languages

Unwatch 1 Star 1 Fork 2

Notifications

Not watching
Be notified when participating or @mentioned.

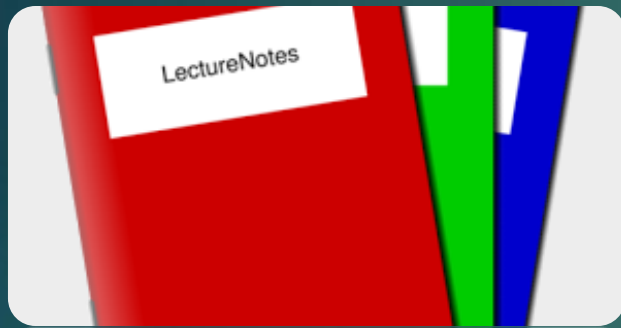
✓ **Watching**
Be notified of all conversations.

Ignoring
Never be notified.

C-Programming [Learning Materials]

4

Er. Shiva K. Shrestha (HoD, Computer Department)
10/3/2018



Syllabus+

C Syllabus

Question Banks

Compilers (Online, C, FORTRAN)



Mini-Project Help

Proposal Format

Project Team Detail

Report Format



Results

First Assessment Result

Final Result

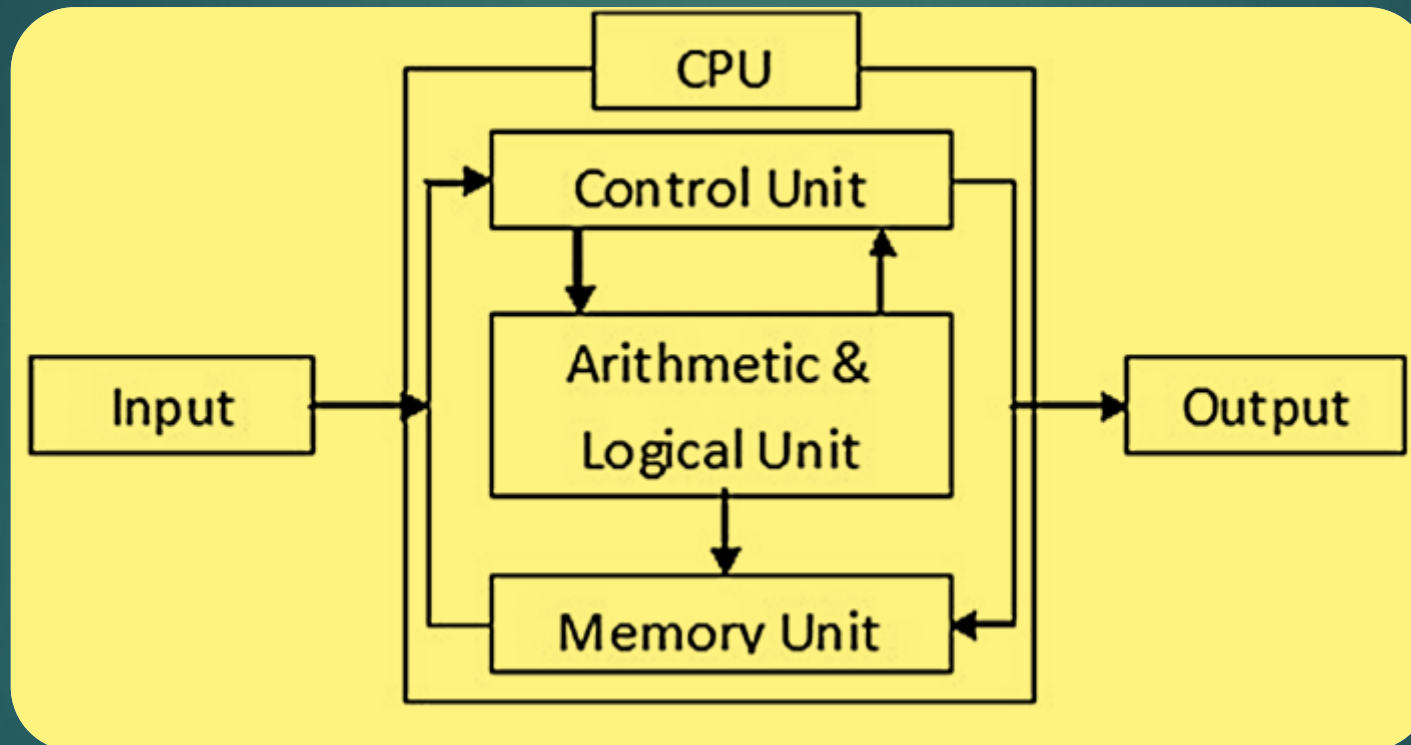
Internal Marks

Practical Marks

Mini-Project Marks

Computer [Intro.]

5



Classification of Computer

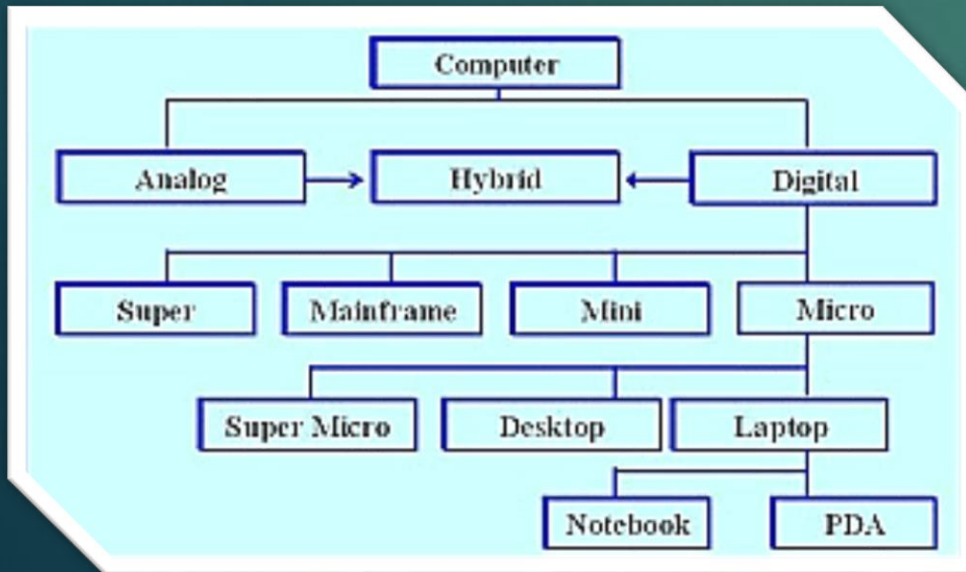
6

On the basis of Working Mode

- ▶ Analog Computer
- ▶ Digital Computer
- ▶ Hybrid Computer

On the basis of Size

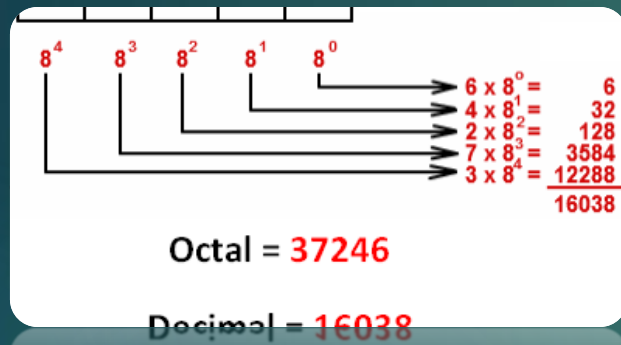
- ▶ Micro Computer
- ▶ Mini Computer
- ▶ Mainframe Computer
- ▶ Super Computer



Number System

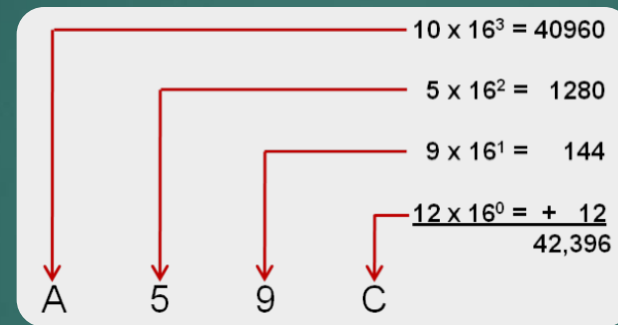
7

Er. Shiva K. Shrestha (HoD, Computer Department)
10/3/2018



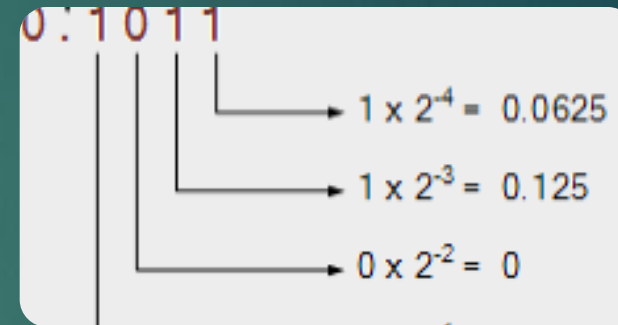
Octal

$2348 = ?_{10}$



Hexadecimal

$B25_{16} = ?_{10}$



Binary

$1101_2 = ?_{10}$

$127_{10} = ?_2$

$0.625_{10} = ?_2$

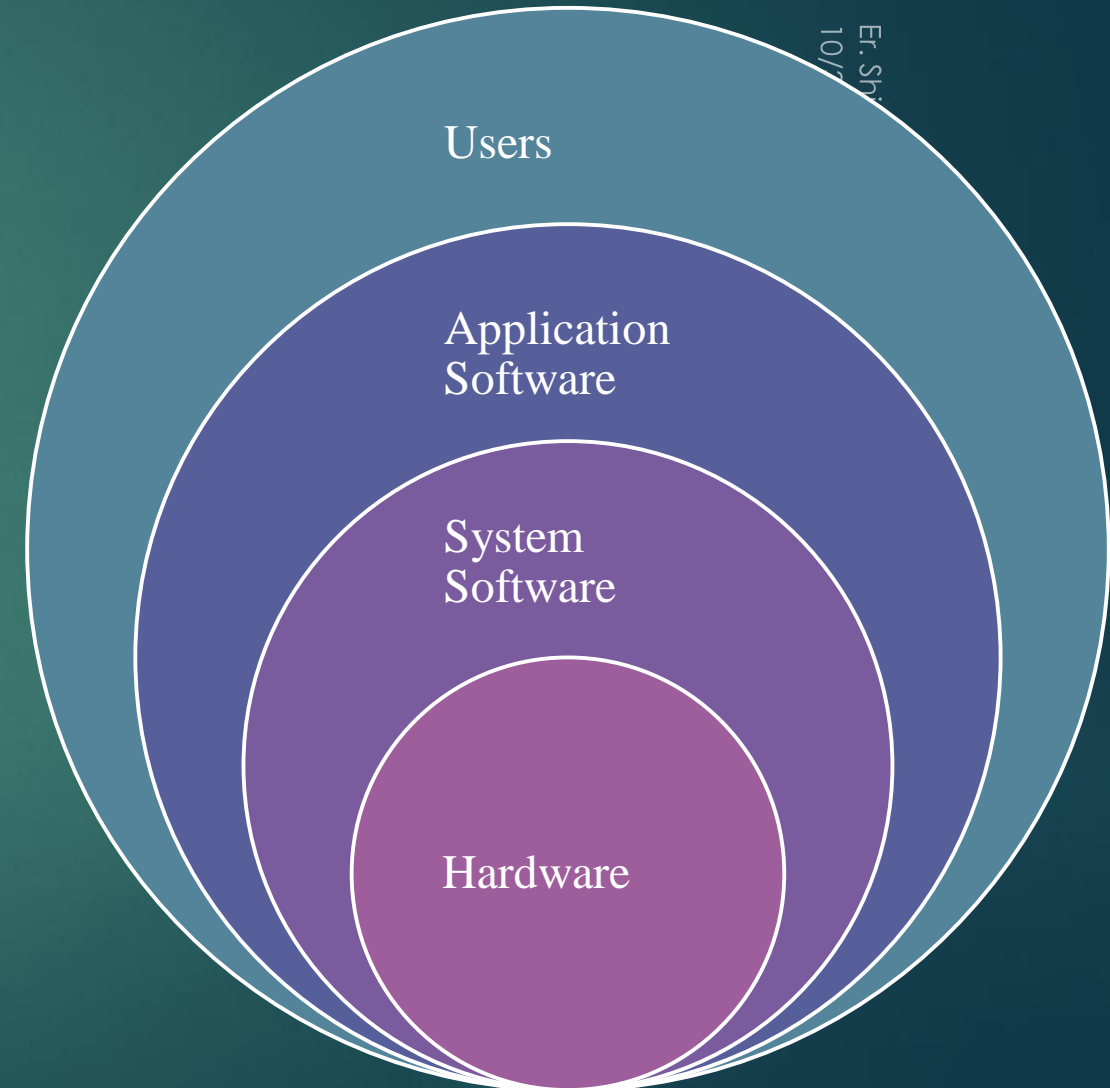
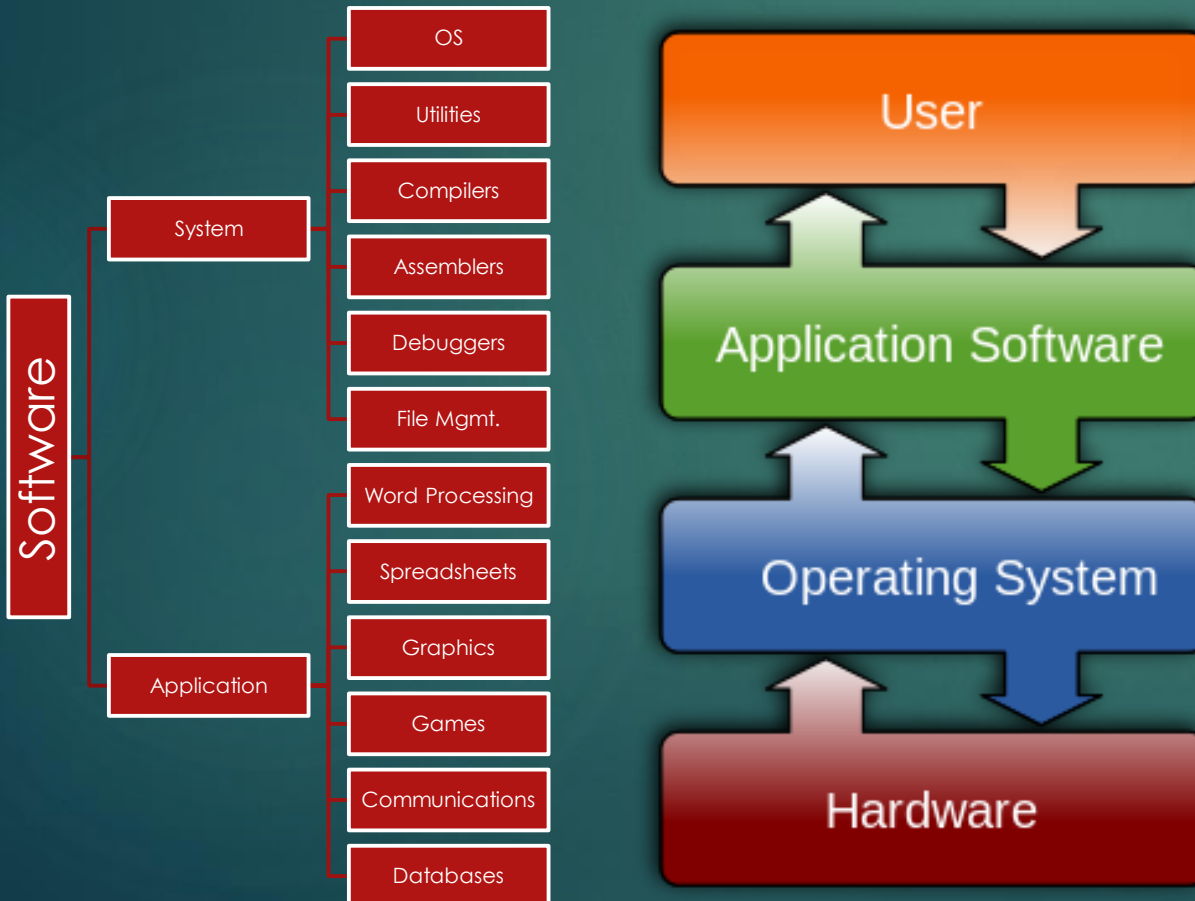
$0.1_{10} = ?_2$

$7.235_{10} = ?_2$

Relationship of Software, Hardware & Users

8

System S/w vs. Application S/w



History & Development of Computers

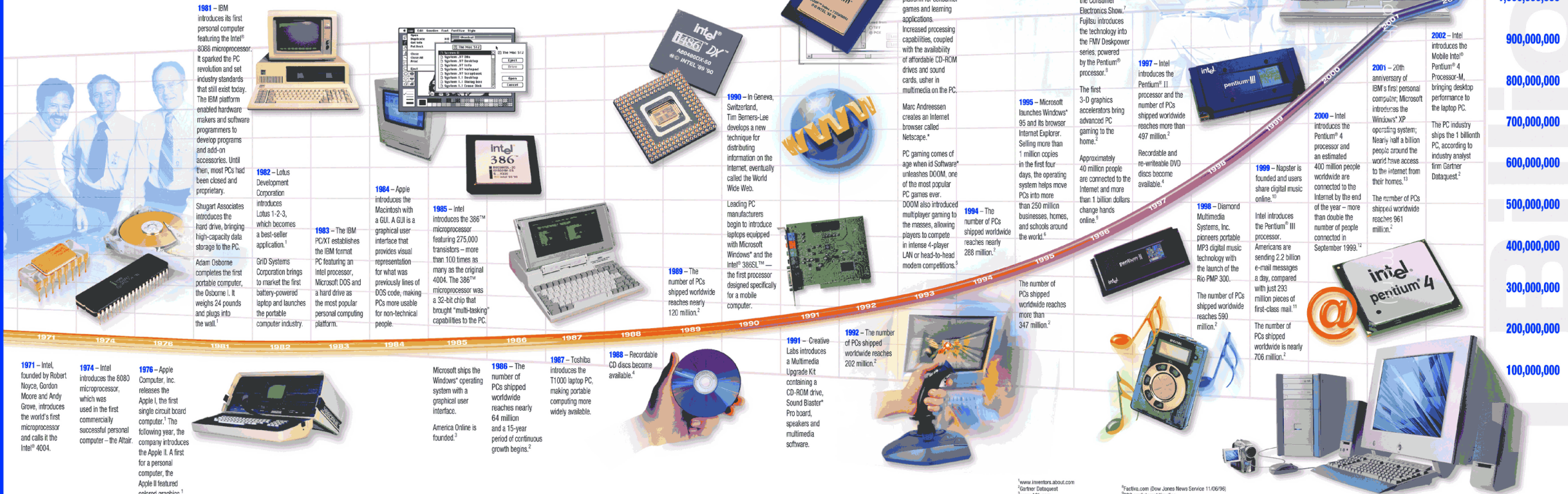
9

intel®

revolution in evolution

Highlights from the Journey to 1 Billion PCs

1,000,000
900,000
800,000
700,000
600,000
500,000
400,000
300,000
200,000
100,000



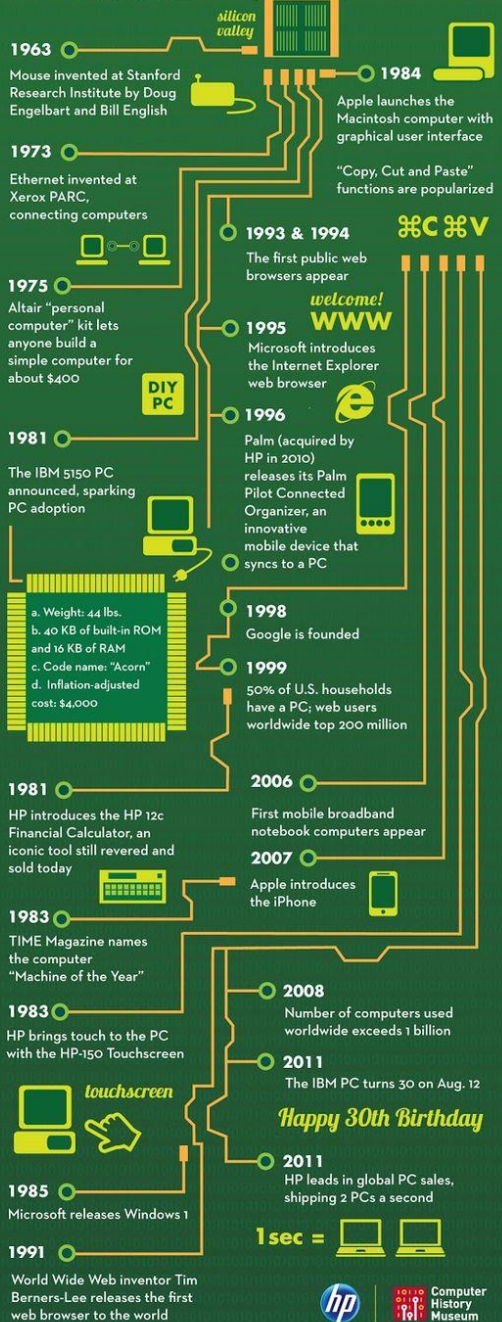
For more information, please visit <http://www.intel.com>

Intel and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Other names and brands may be claimed as the property of others.

¹www.inventors.about.com
²Gartner Dataquest
³www.AOL.com
⁴www.discuss.com/ind
⁵www.Stuffware.com
⁶www.microsoft.com
⁷Consumer Electronics Association

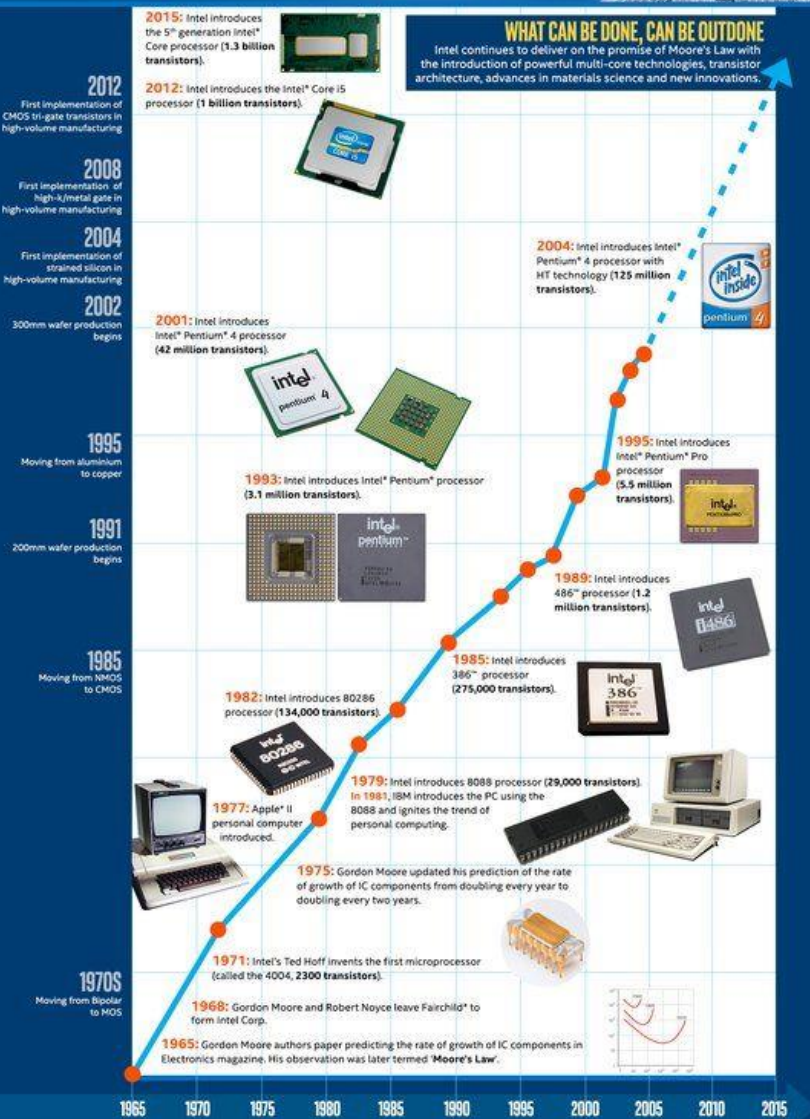
⁸Factiva.com (Dow Jones News Service 11/06/96)
⁹PBS.org/internetimeline
¹⁰www.napster.com
¹¹US News & World Report, 3/22/99
¹²Nua Internet Surveys, 12/00
¹³NelsonA/Batings, "Half Billion Can Surf From Home," Reuters 3/02

Great Moments in PC HISTORY



MOORE'S LAW TIMELINE

Moore's Law – the observation that computing dramatically decreases in cost at a regular pace – is short-hand for rapid technological change. Over the past 50 years, it has ushered in the dawn of the personalization of technology and enabled new experiences through the integration of technology into almost all aspects of our lives.



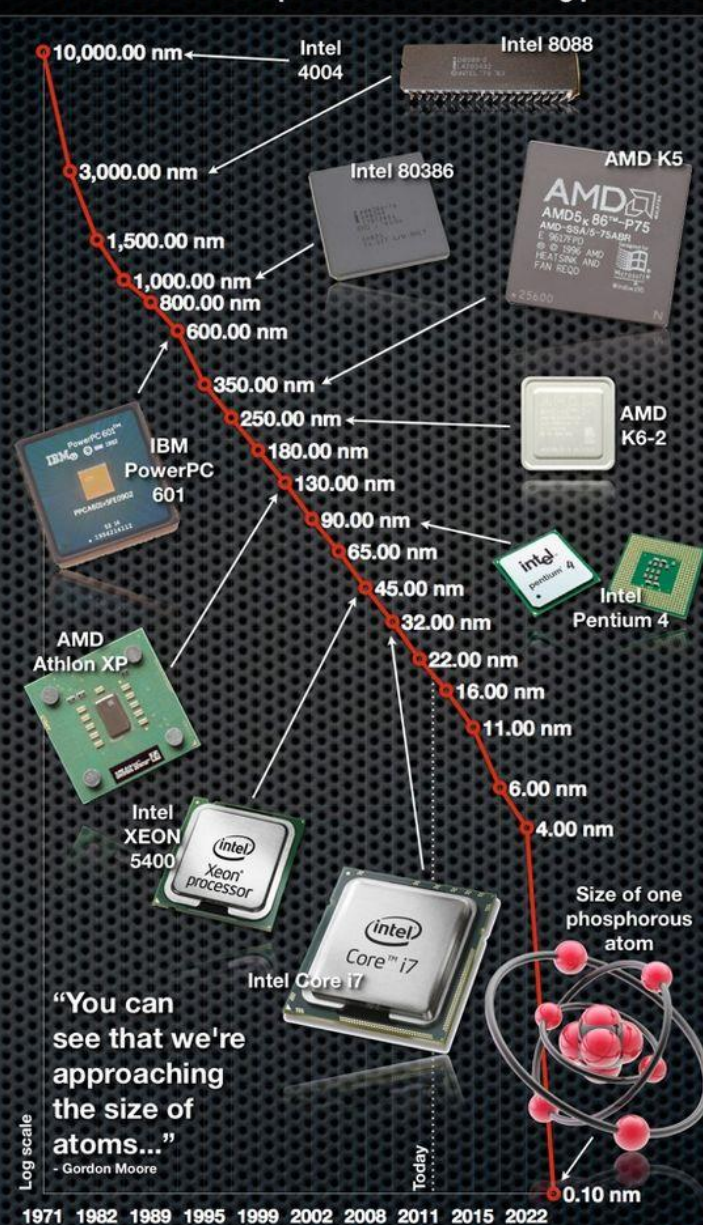
For more information, please visit intel.com.

Intel, Pentium, Core, Intel386, Intel486, the Intel logo and the Intel Inside logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. *Other names and brands may be claimed as the property of others.



How small can a transistor be?

The evolution of microprocessor manufacturing processes



Data source: Wikipedia
Graphics from Intel, Shutterstock, and Wikipedia

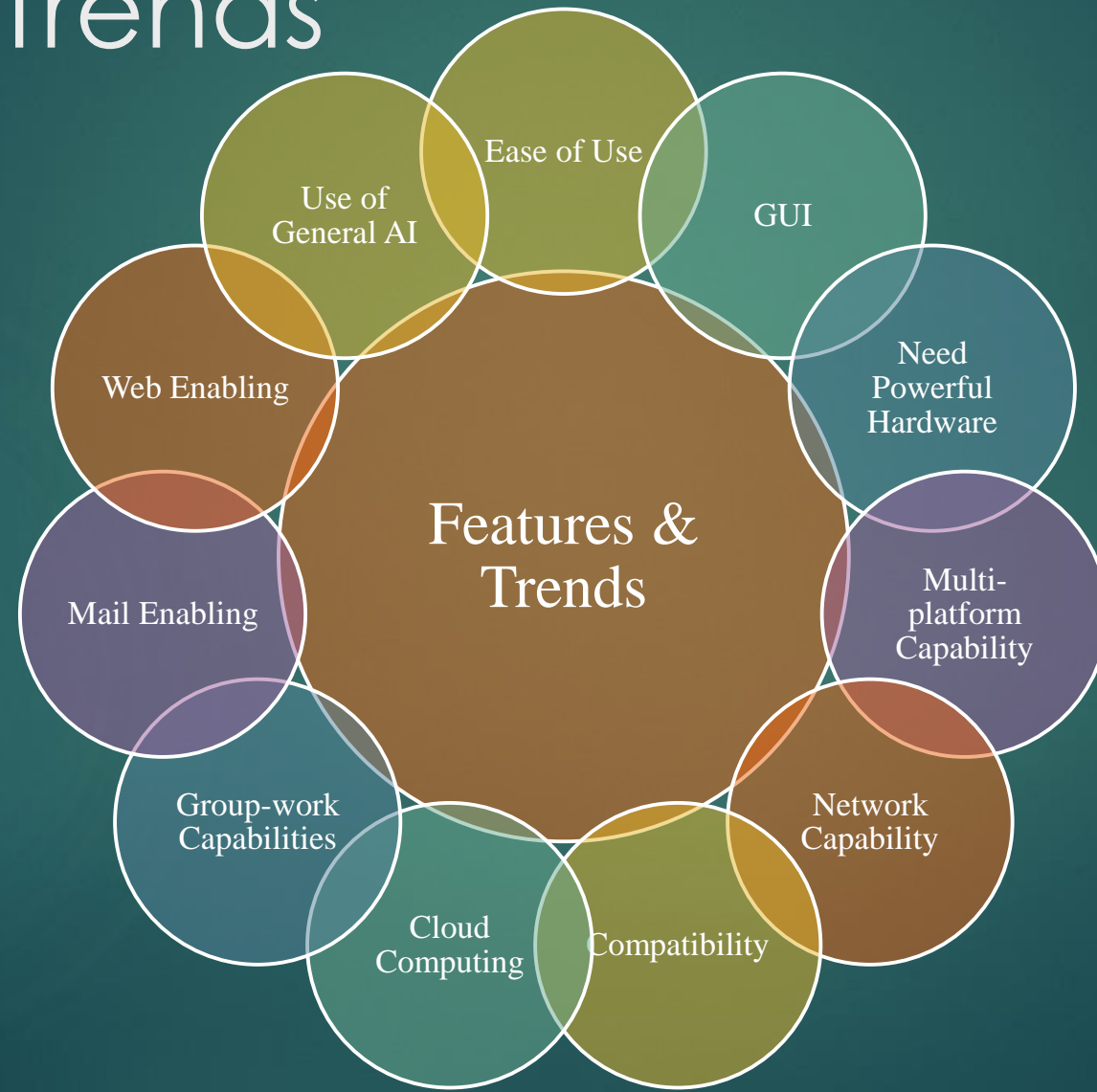
www.pingdom.com

10

Er. Shiva K. Shrestha (HoD, Computer Department)
10/3/2018

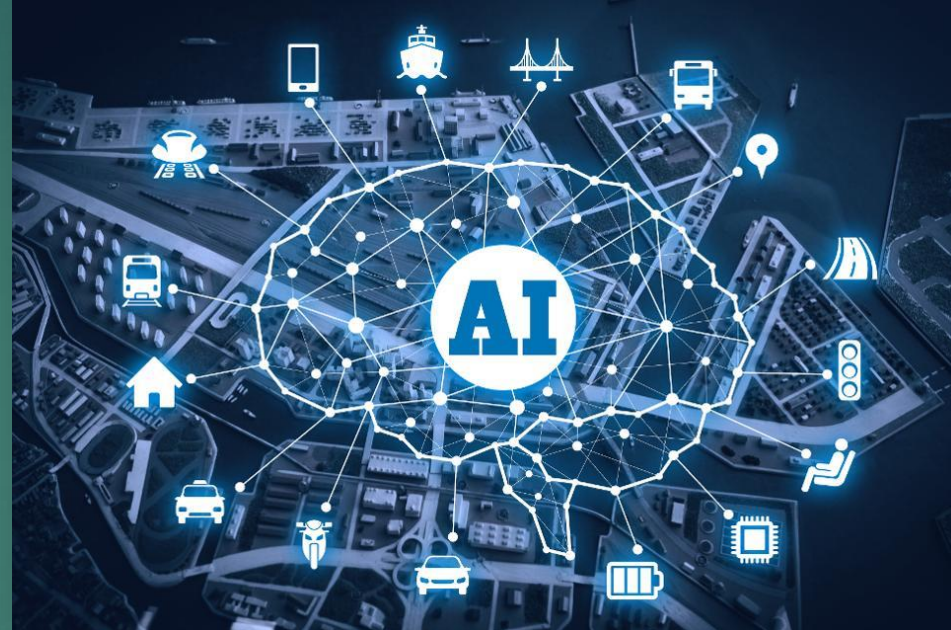
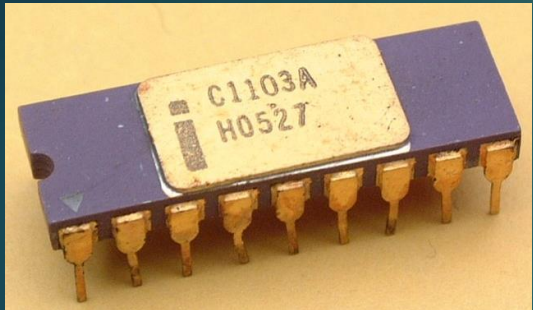
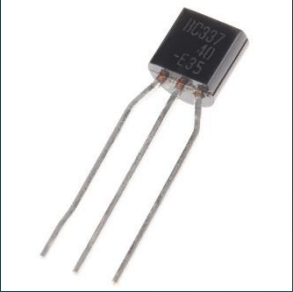
General Software Features & Recent Trends

11



Generation of Computer

12



1st Gen

- Vacuum Tubes
- 1945-55

2nd Gen

- Transistors
- 1955-65

3rd Gen

- IC
- 1965-1980

4th Gen

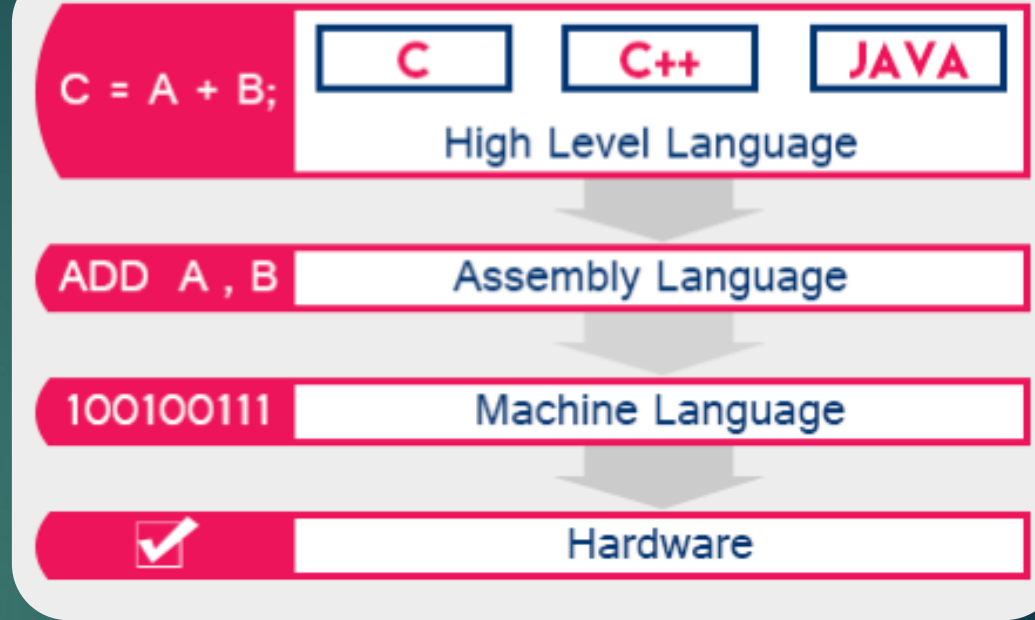
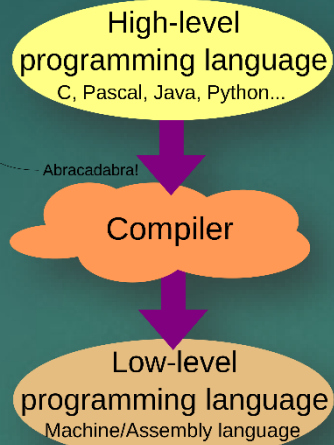
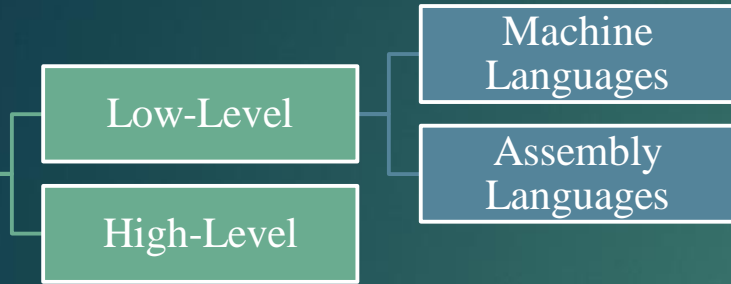
- μ P
- 1980

5th Gen

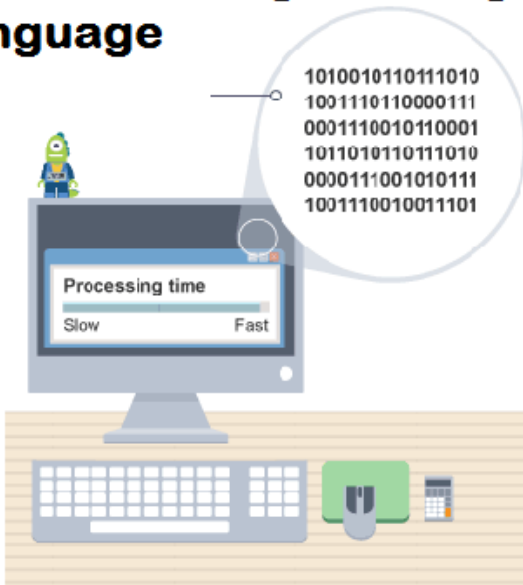
- AI
- Present & Beyond

Programming Languages

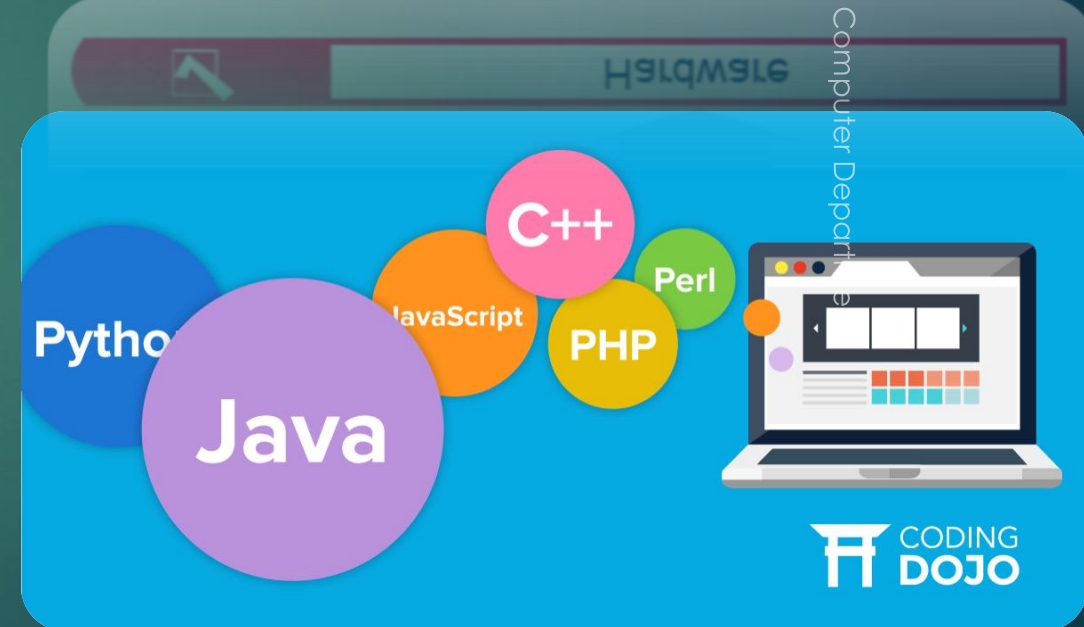
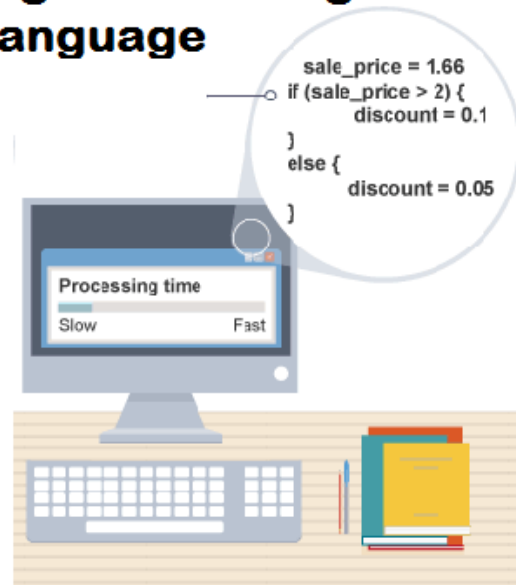
Programming Language



Low Level Programming Language



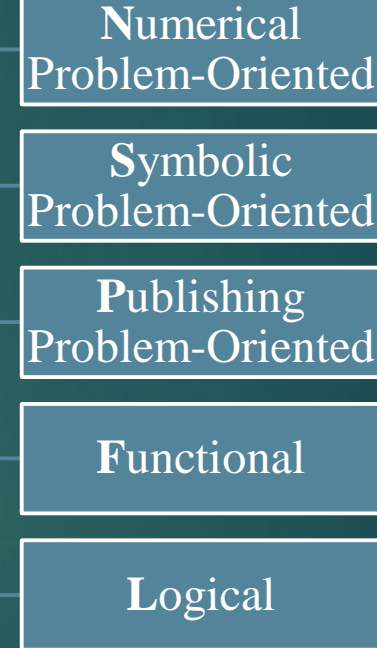
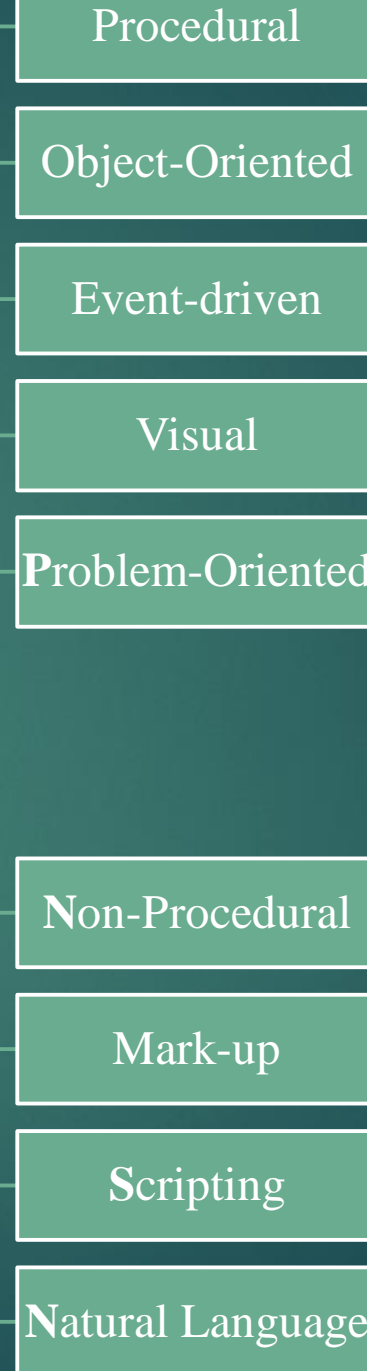
High Level Programming Language



High Level Language Categorization

- ▶ Procedural
- ▶ Object-Oriented
- ▶ Event-driven
- ▶ Visual
- ▶ Problem-Oriented
- ▶ Non-Procedural
- ▶ Mark-up
- ▶ Scripting
- ▶ Natural Language

High Level Language



High Level vs. Low Level

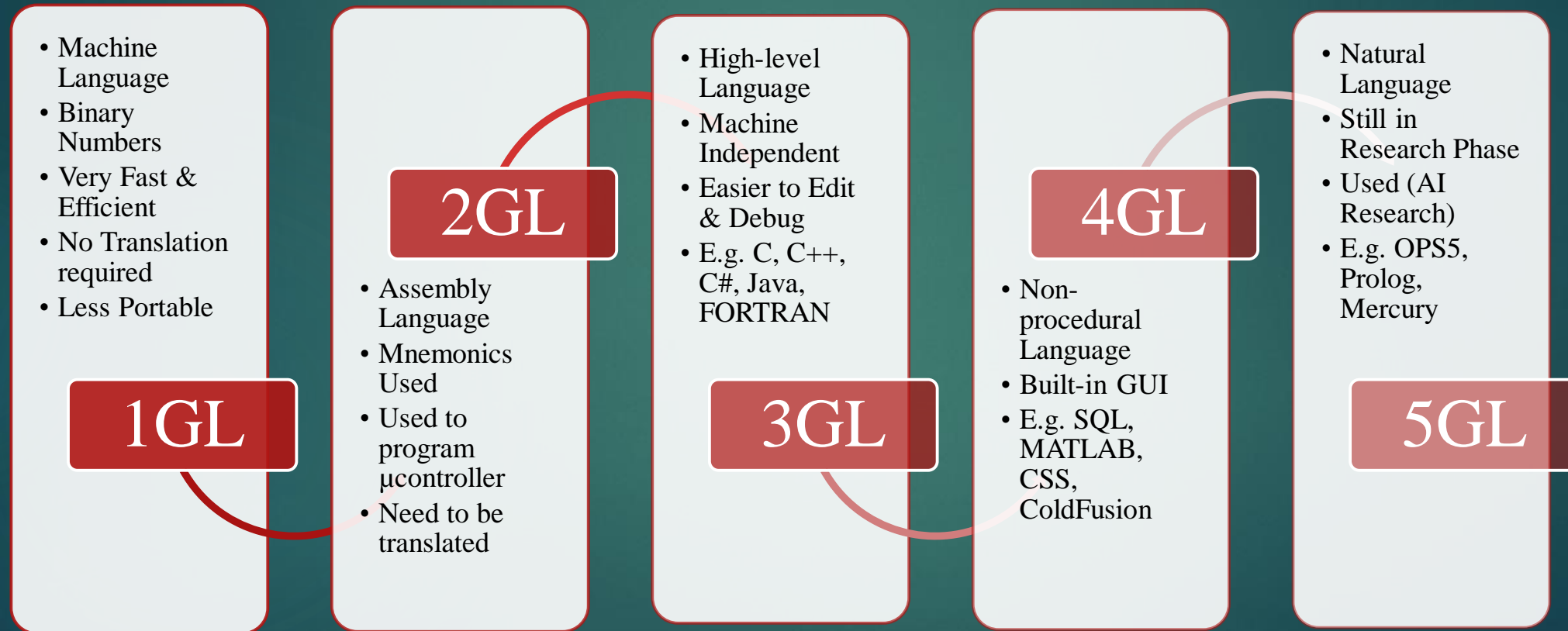
15

Er. Shiva K. Shrestha (HoD, Computer Department)
10/3/2018

High Level Language	Low Level Language
These are Interpreted	Direct memory management
They have open classes and message-style methods which are known as Dynamic constructs	Hardware has extremely little abstraction which is actually close to having none.
Poor performance	Much fast than high level
Codes are Concise	Statements correspond directly to clock cycles
Flexible syntax and easy to read	Superb performance but hard to write
Is object oriented and functional	Few support and hard to learn
Large community	

Generation of Programming Languages

16



Generation Computer & Programming Language

17

Er. Shiva K. Shrestha (HoD, Computer Department)
10/3/2018

Generation of Computer

1st Gen	2nd Gen	3rd Gen	4th Gen	5th Gen
<ul style="list-style-type: none">• Vacuum Tubes• 1945-55	<ul style="list-style-type: none">• Transistors• 1955-65	<ul style="list-style-type: none">• IC• 1965-1980	<ul style="list-style-type: none">• μP• 1980	<ul style="list-style-type: none">• AI• Present & Beyond

- ▶ **History** of Computer Development – **Generation of Computer**
- ▶ Characterized by **major technological development** that fundamentally changed the way computers operate, resulting in increasingly **smaller**, **cheaper**, more **powerful**, more **efficient**, and **reliable** devices.

Generation of Programming Languages

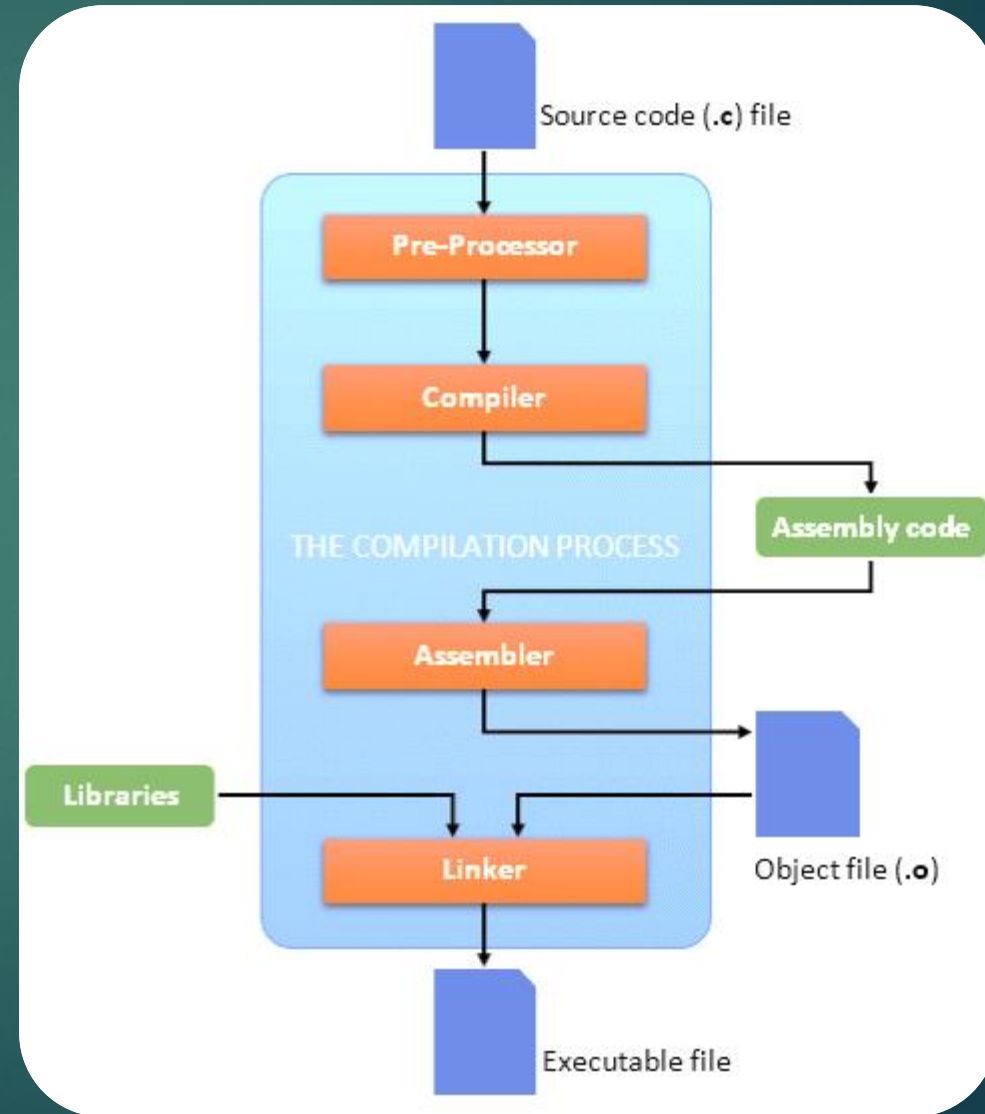
- ▶ Programming language is a **set of written symbols** that instructs the computer hardware to perform specific tasks
- ▶ Programming languages are classified into five generations depending upon the **power** and **flexibility** of programming styles.



Compilation Process

18

1. Pre-processing
2. Compilation
3. Assembling &
4. Linking



Interpreter vs. Compiler

19

Er. Shiva K. Shrestha (HoD, Computer Department)
10/3/2018

Interpreter	Compiler
Translates program - one statement at a time.	Scans the entire program and translates it as a whole into machine code.
It takes less amount of time to analyze the source code but the overall execution time is slower.	It takes large amount of time to analyze the source code but the overall execution time is comparatively faster.
No intermediate object code is generated, hence are memory efficient.	Generates intermediate object code which further requires linking, hence requires more memory.
Continues translating the program until the first error is met, in which case it stops. Hence debugging is easy.	It generates the error message only after scanning the whole program. Hence debugging is comparatively hard.
Programming language like Python, Ruby use interpreters.	Programming language like C, C++ use compilers.

Q/A?

20

Thank You!

Er. Shiva K. Shrestha

computer.khwopa@gmail.com