

CSC005 – Overview Of Computer Science



Course Description

- Introduction to fundamental concepts in computer hardware and software. Exploration of the history and evolution of computing, and foundational areas of current computer science research. Algorithms, program development, and problem solving. Elements and use of a high-level programming language.
- *Prerequisites:* None

Text

Required Text:

Computer Science Illuminated, Second Edition

Nell Dale, Ph.D., University of Texas, Austin, John Lewis,
Villanova University

ISBN 13: 9780763726263

ISBN 10: 0763726265

Price: \$81.95 (Sugg. US List)

Cover: Paperback

Pages: 699

Copyright: 2004

Grading

- Several **assignments**
- **mid-term** and **end-term**
- Class **participation**
- Final **project** or **paper**
- **No make-up** test or **extended deadlines**

Point Allocation

Assignments 1-3:	5% each
Final Project:	30%
Mid-Term:	25%
End-Term:	25%
Participation:	5%

Attendance

- **Not Mandatory**, but...
- ...you'll probably **fail!**
- **Participation** is very important
- **Let me know** if you can't make it

Course Schedule

Chapter 1: The Big Picture

Chapter 2: Binary Values and Number Systems

Chapter 3: Data Representation

Chapter 4: Gates and Circuits

Chapter 5: Computing Components

Chapter 6: Problem Solving and Algorithm Design

Chapter 7: Low-Level Programming Languages

Chapter 8: High-Level Programming Languages

Chapter 9: Abstract Data Types and Algorithms

Chapter 10: Operating Systems

Chapter 11: File Systems and Directories

Chapter 12: Information Systems

Chapter 13: Artificial Intelligence

Chapter 14: Simulation and Other Applications

Chapter 15: Networks

Chapter 16: The World Wide Web

Chapter 17: Limitations of Computing

Slides, Links & News

- www.cs.hofstra.edu/~cscvjc/Fall06

Class Rules

- Assignments are to be completed individually
- Academic **honesty** taken very seriously
- Be selfless! – You are part of a community

Chapter 1

- Introduction – The Big Picture

Computing Systems

Computing systems are dynamic!

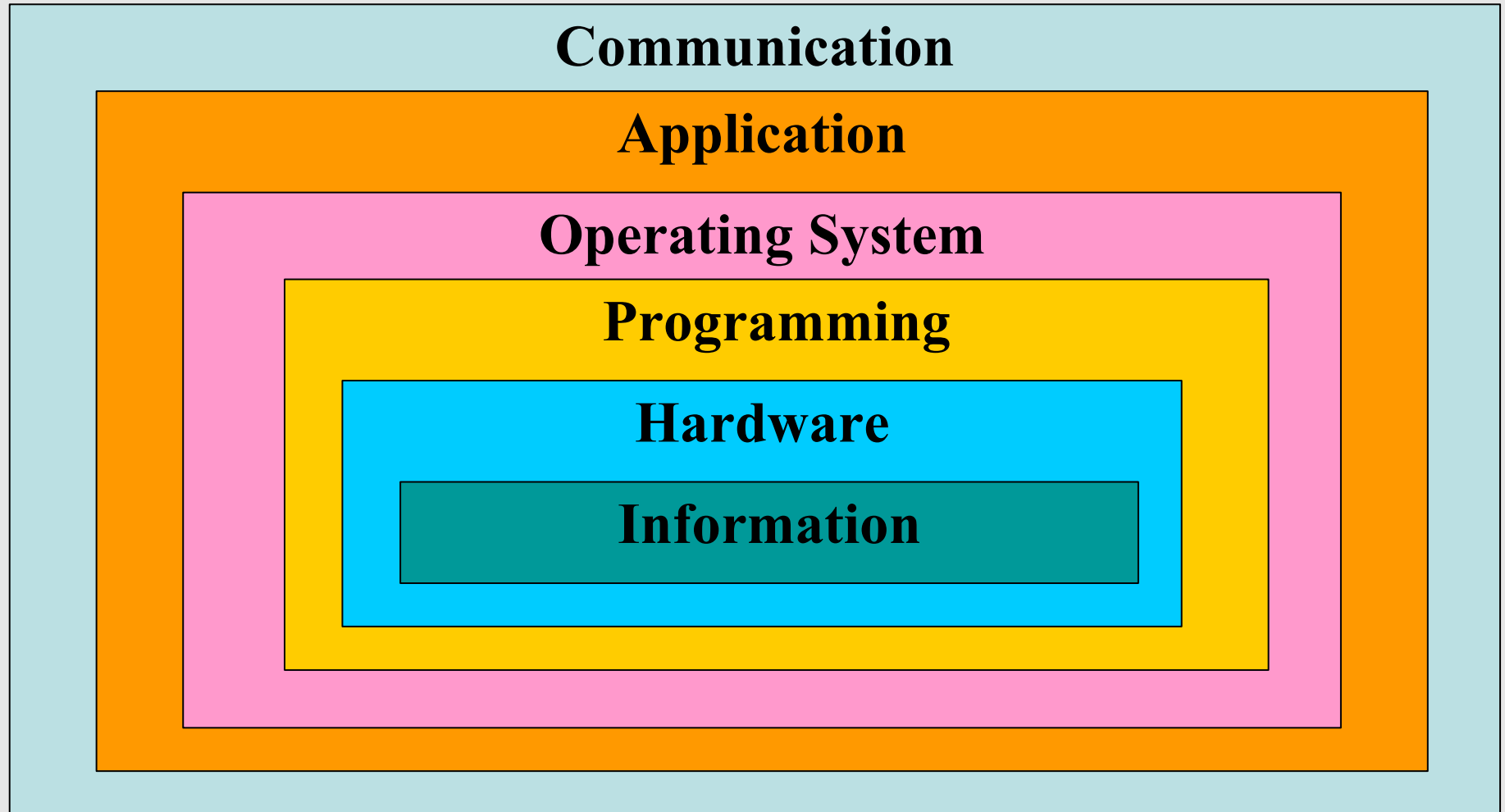
*What is the difference between **hardware** and **software**?*

Computing Systems

Hardware The physical elements of a computing system (printer, circuit boards, wires, keyboard...)

Software The programs that provide the instructions for a computer to execute

Layers of a Computing System



Abstraction

Abstraction A mental model that removes complex details

This is a key concept. Abstraction will reappear throughout the text – be sure to understand it!

Early History of Computing

Abacus

An early device to record numeric values

Blaise Pascal

Mechanical device to add, subtract, divide & multiply

Joseph Jacquard

Jacquard's Loom, the punched card

Charles Babbage

Analytical Engine

Early History of Computing

Ada Lovelace

First Programmer, the loop

Alan Turing

Turing Machine, Artificial Intelligence Testing

Harvard Mark I, ENIAC, UNIVAC I

Early computers launch new era in mathematics, physics, engineering and economics

First Generation Hardware (1951-1959)

Vacuum Tubes

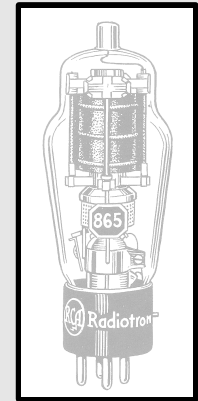
Large, not very reliable, generated a lot of heat

Magnetic Drum

Memory device that rotated under a read/write head

Card Readers → Magnetic Tape Drives

Sequential auxiliary storage devices



Second Generation Hardware (1959-1965)



Transistor

Replaced vacuum tube, fast, small, durable, cheap

Magnetic Cores

Replaced magnetic drums, information available instantly

Magnetic Disks

Replaced magnetic tape, data can be accessed directly

Third Generation Hardware (1965-1971)

Integrated Circuits

Replaced circuit boards, smaller, cheaper, faster, more reliable.

Transistors

Now used for memory construction

Terminal

An input/output device with a keyboard and screen

Fourth Generation Hardware (1971-?)

Large-scale Integration

Great advances in chip technology

PCs, the Commercial Market, Workstations

Personal Computers were developed as new companies like Apple and Atari came into being. Workstations emerged.

Parallel Computing and Networking

Parallel Computing

Computers rely on interconnected central processing units that increase processing speed.

Networking

With the Ethernet small computers could be connected and share resources. A file server connected PCs in the late 1980s.

ARPANET and LANs → Internet

First Generation Software (1951-1959)

Machine Language

Computer programs were written in binary (1s and 0s)

Assembly Languages and translators

Programs were written in artificial programming languages and were then translated into machine language

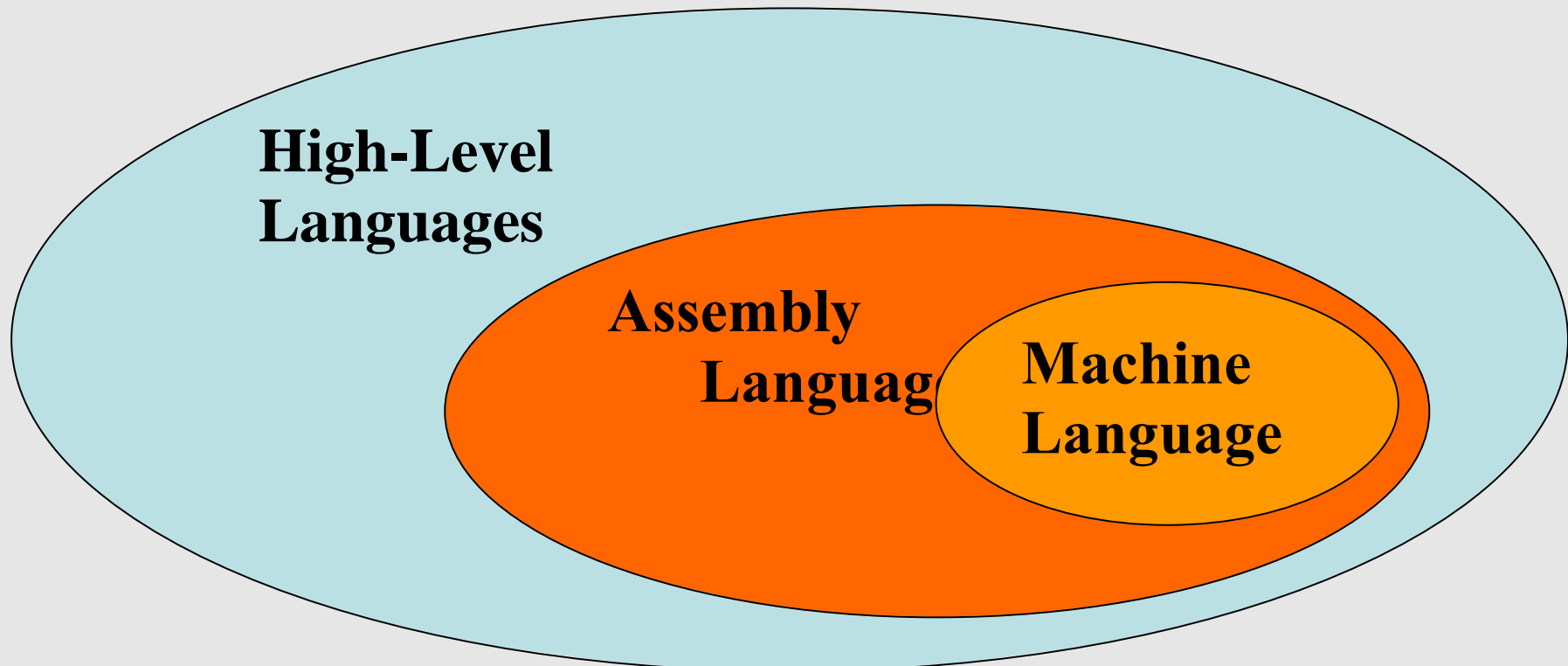
Programmer Changes

Programmers divide into application programmers and systems programmers

Second Generation Software (1959-1965)

High Level Languages

Use English-like statements and make programming easier. Fortran, COBOL, Lisp are examples.



Third Generation Software (1965-1971)

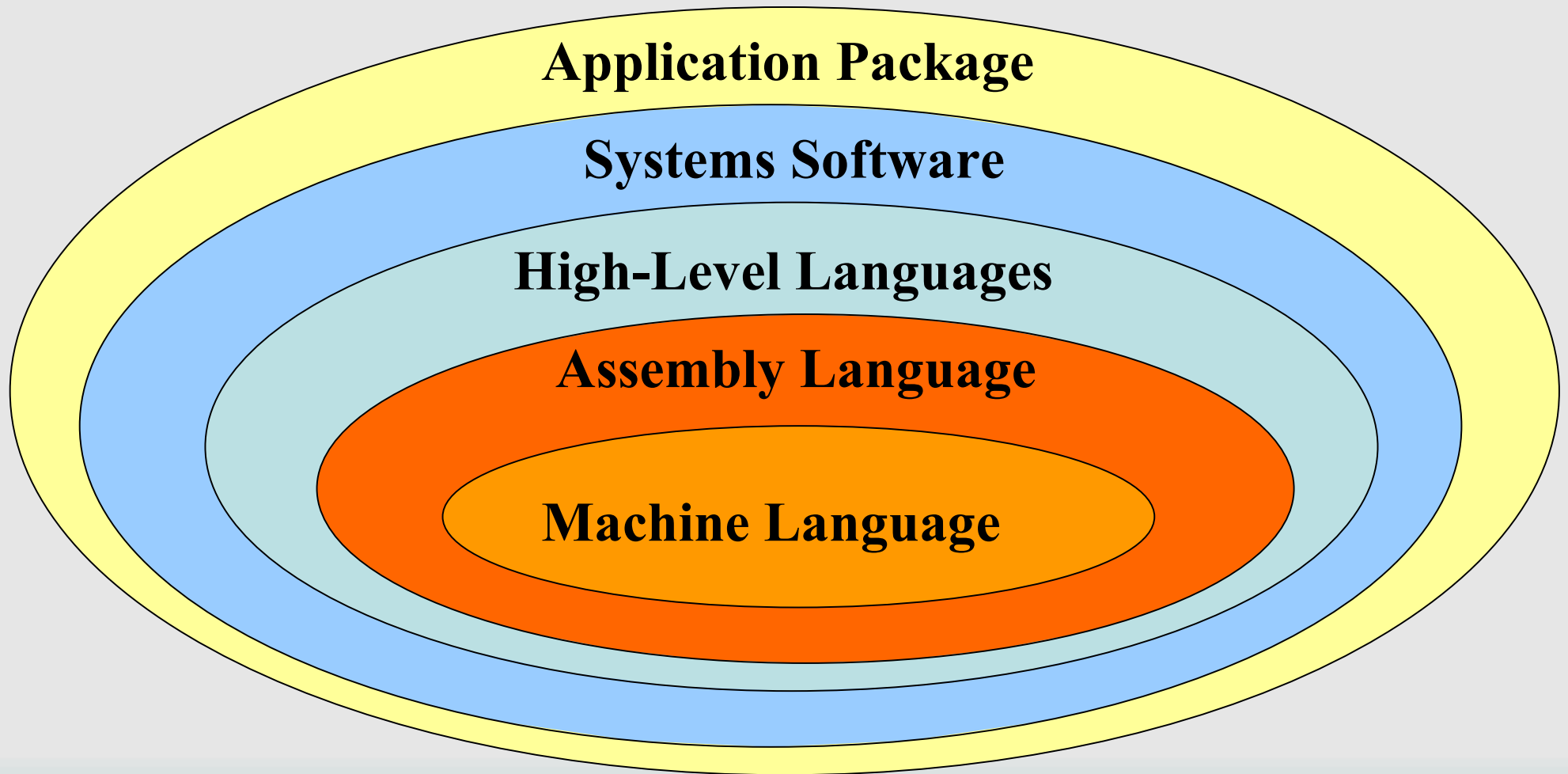
- **Systems Software**

- utility programs,
- language translators,
- and the operating system, which decides which programs to run and when.

- **Separation between Users and Hardware**

Computer programmers began to write programs to be used by people who did not know how to program

Third Generation Software (1965-1971)



Fourth Generation Software (1971-1989)

Structured Programming

Pascal, C, C++

New Application Software for Users

Spreadsheets, word processors, database management systems

Fifth Generation Software (1990-present)

Microsoft

The Windows operating system, and other Microsoft application programs dominate the market

Object-Oriented Design

Based on a hierarchy of data objects (i.e. Java)

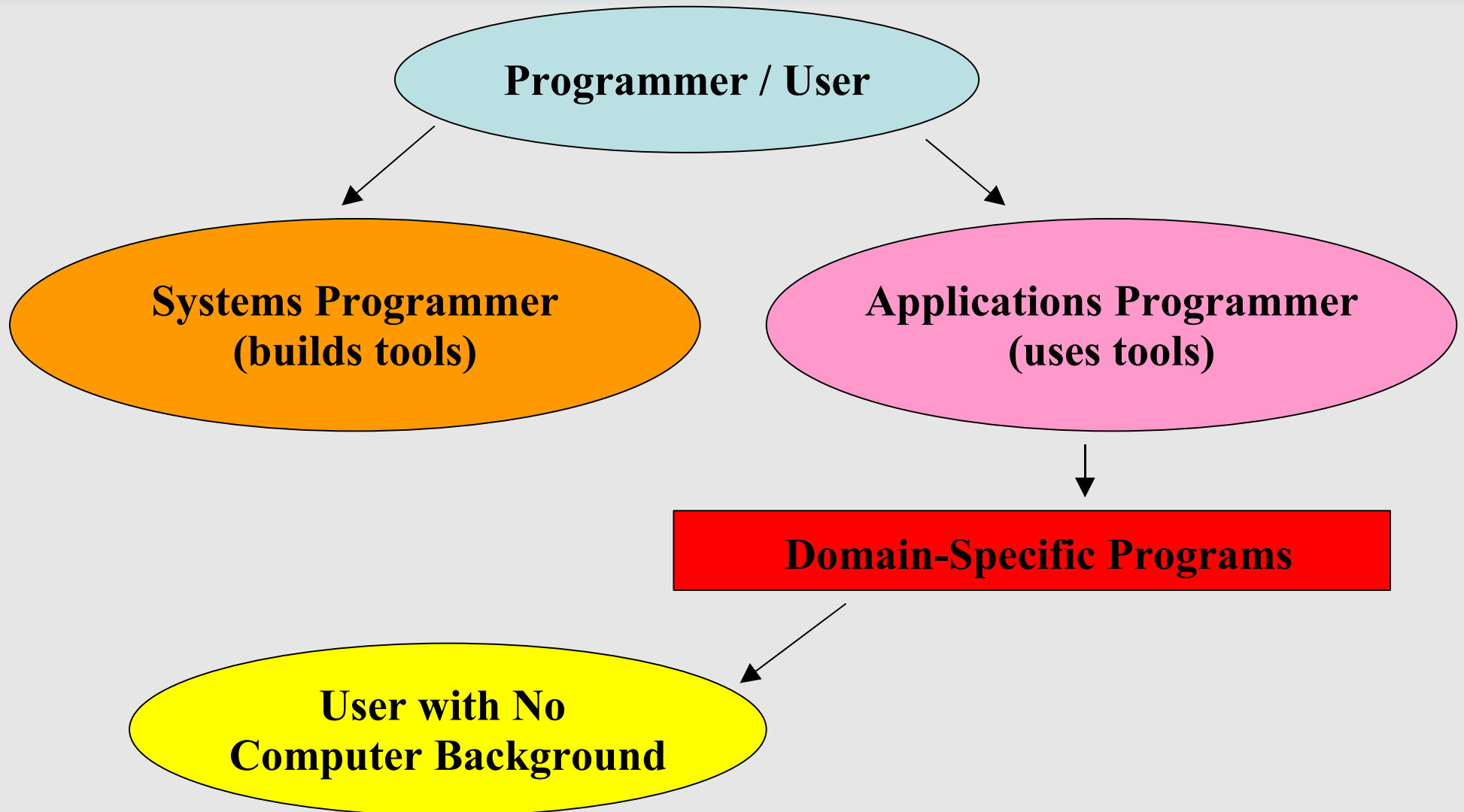
World Wide Web

Allows easy global communication through the Internet

New Users

Today's user needs no computer knowledge

Computing as a Tool



Computing as a Discipline

- ***What can be (efficiently) automated?***
- **Four Necessary Skills**
 1. Algorithmic Thinking
 2. Representation
 3. Programming
 4. Design

Computing as a Discipline

What do you think?

Is Computer Science a mathematical, scientific, or engineering discipline?

Systems Areas of Computer Science

- **Algorithms and Data Structures**
- **Programming Languages**
- **Architecture**
- **Operating Systems**
- **Software Methodology and Engineering**
- **Human-Computer Communication**

Application Areas of Computer Science

- **Numerical and Symbolic Computation**
- **Databases and Information Retrieval**
- **Artificial Intelligence and Robotics**
- **Graphics**
- **Organizational Informatics**
- **Bioinformatics**

Social Networking

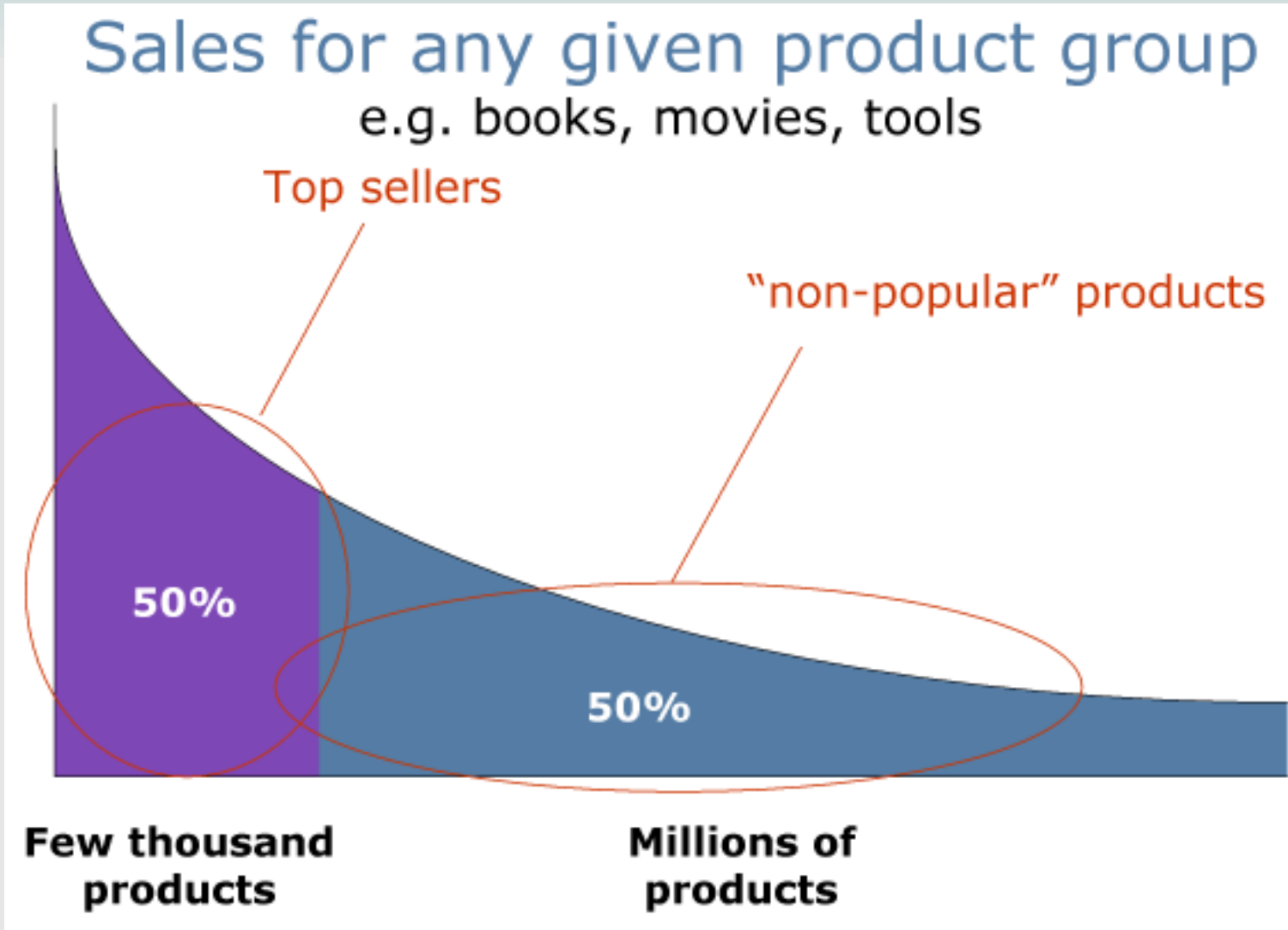
- **Social Networking** - Bringing People Together Through Collaboration, Content Syndication, Folksonomy, Web Service, Web 2.0
- Jabber/XMPP - Open Alternative To Instant Messaging
- Tools - Collaboration With Wiki's, Joomla, del.icio.us, Protopage, GooglePages
- Social Sites – Flickr, MySpace, YouTube

The Long Tail

- Coined by Chris Anderson in Oct, 2004 Wired article
- XY: Popularity vs Inventory
- Total volume of low popularity items exceeds the volume of high popularity items
- Key factor: cost of inventory storage and distribution
- It can be economically viable to sell relatively unpopular products



The Long Tail



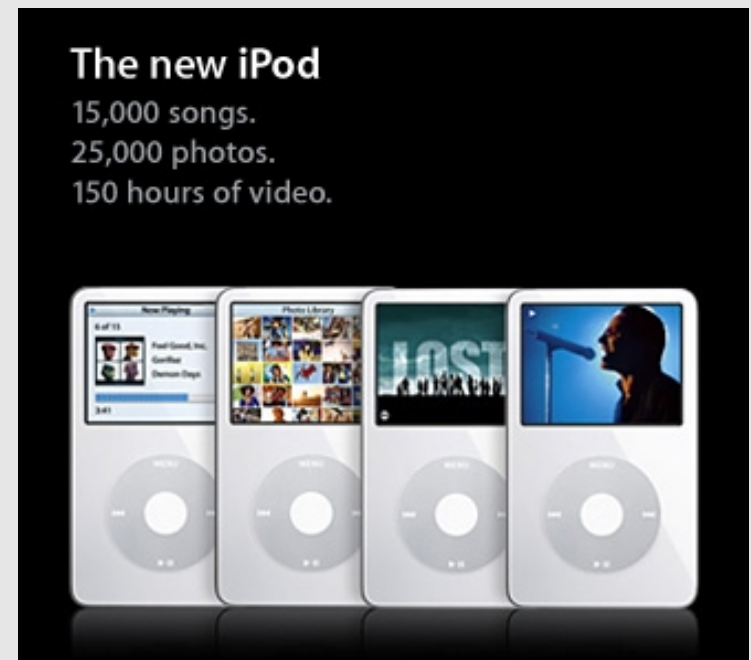
- Coined by Chris Anderson in Wired

Making A Difference

[Apple Advertisement, 10/13] “It’s unfolded before your eyes. The revolution that is iPod first took the music scene by storm. Further spiced things up with full-color photos. Added a full complement of podcasts to the mix.

And now iPod has turned the world topsy-turvy once again with video, letting you carry up to 150 hours of video wherever you go. Imagine: With iPod, you can play the DJ one minute. Rock with the latest Madonna or U2 music videos the next. Then get lost with “Lost”—or any of the other TV shows or short films now available for purchase and download from the iTunes Music Store. “

The Long Tail is becoming reality!!!



Have A Good Week



licensed under Attribution-NonCommercial-ShareAlike 2.0 Germany | Ludwig Gatzke | <http://flickr.com/photos/stabilo-boss/>