ITI 1120 Fall 2014
Introduction to Computing I

- Professor: Diana Inkpen
- Office: STE 5015
- E-mail: **diana.inkpen@uottawa.ca**
  - E-mail is the preferred way to contact me.
- Phone: 562-5800 x6711
- Office hours: to be determined.

On-line Course Material

- Course web page:

  `http://www.eecs.uottawa.ca/~diana/iti1120`

  `Lecture notes, lab notes`

- Virtual Campus - **Blackboard LEARN**:

  On-line course communication tools

  - More notes, examples, solutions
  - Assignment submission
  - Grades

- Access:

  `https://maestro.uottawa.ca/index.asp?LANG=EN`

  - Choose **Blackboard LEARN Login**

References

Course notes:

- PDF file available on course web page
- Print and bring the notes to every class!

Text (recommended, but not required)

- Available at the SFUO "Agora" bookstore, 145 Besserer St. (www.agorabookstore.ca)
- Customized textbook for this course, with only the necessary chapters from the textbook: "Introduction to Java Programming," by Y.D. Liang. (cost ~ $60)
- Used copy of: "Java Programming: From The Ground Up", by Ralph Bravaco and Shai Simonson, Mcgraw Hill Higher Education, 2009.
- Alternate, optional references:
  - "A First Book of Java" by G. Bronson
  - "Introduction to Java Programming" 4th Edition by Y. D. Liang

Labs

- Scheduled labs will be in groups of no more than 30 students, with help available from a Teaching Assistant.
- You will be in a room with computers available for each student.
- Involves practical work with the Java programming language.
- You should have been assigned to one of the available lab sections, and you must attend that session each week.
  - Your lab section assignment is available via the Registration Navigator
- The STE 0110 general lab is available at other times.

Labs

- Regular labs will start on the second week
- Please read the lab manual before your first lab.
- The first lab:
  - There will be an introduction to basic concepts:
    - PCs, Windows, internet browsers
      - Includes information peculiar to our lab environment: network file storage
    - e-mail
    - virus detection,
  - How to use the Virtual Campus to submit assignments
  - Use of the "Dr. Java" environment.

Assignments

- There will be 5 assignments during the term.
- Most assignments will involve programming in Java; some will involve written work.
- Read the instructions on the course web site about how to submit assignments.
  - These instructions must be followed, or marks will be deducted.
  - A practice assignment is posted on the virtual campus, to try the assignment submission system.

Midterm Examination

- Date:
  - After reading week, TBA

- The midterm will be an 90-minute closed-book test.

Final Examination

- To be scheduled by the University Registrar
  - The exam could be scheduled by the faculty during the exam period
  - Check the university web site later for the date, time, and location.
- The exam will be 3 hours, closed-book.

Determination of Final Grade

- Average of 5 assignments: 25% (5% each)
- Midterm: 20%
- Final examination: 50%
- Lab exercises and participation (5%)
  (there are 11 labs, 0.5% each lab, therefore one lab can be missed without penalty).
- Bonus points for class participation.

- You must get at least 50% from the midterm and final exam, to pass the course.
  - Otherwise, the score will be converted to a percentage, and your final grade will be E or F.

Things to do right away

1. Read the course web pages
2. Read the course lab manual, available on-line
   - Also read the course SITE lab introductory guide.
3. Print the course lecture notes.
4. Buy your textbook, or use one of the on-line references.
5. Obtain your SITE computer account for the Computer Laboratory.
6. If you want to work on your own computer, get the Java development kit (JDK) installed on your computer.

Course Policies

- See course web site for official descriptions

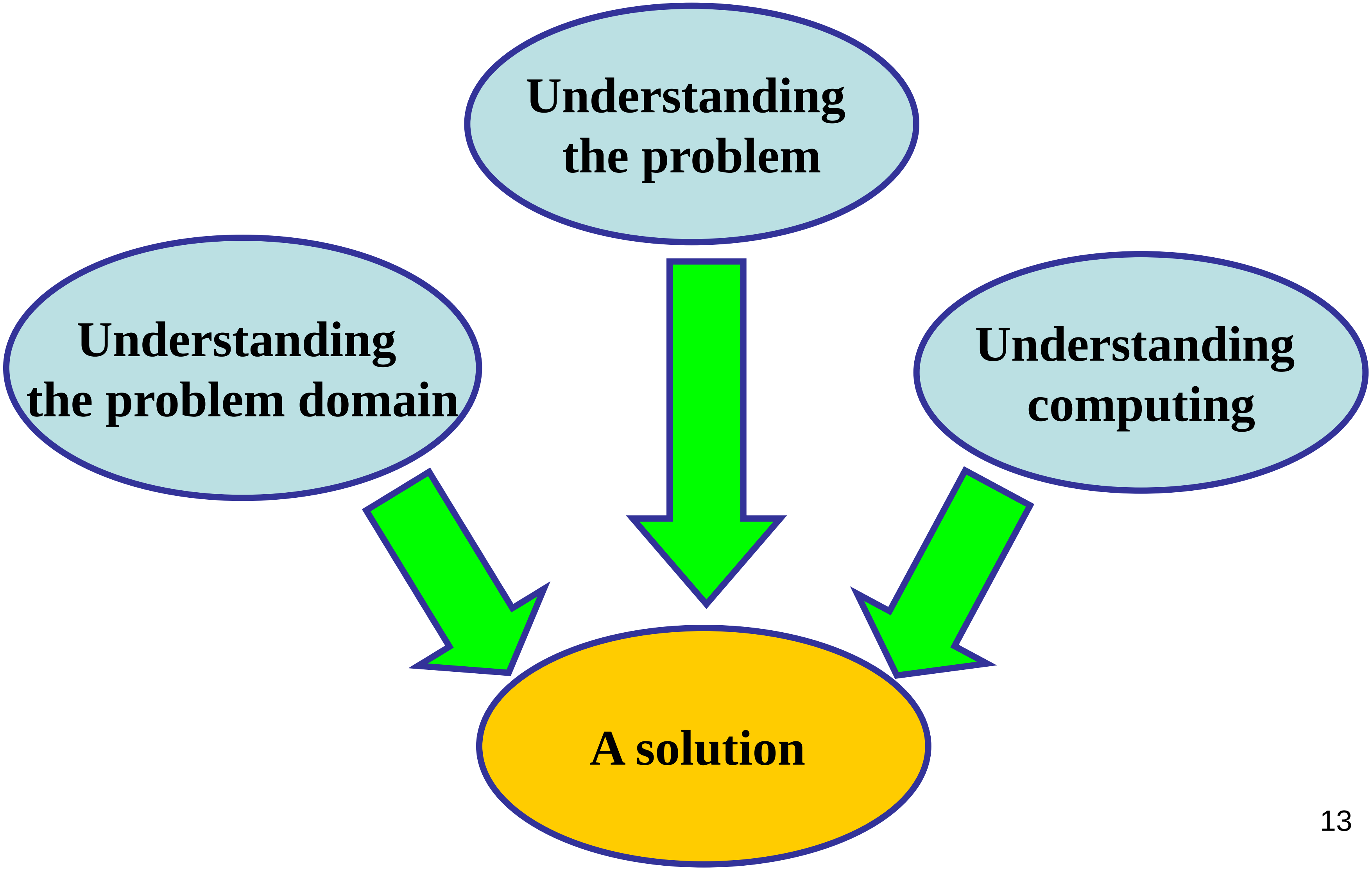
- Missed/late assignments: mark of zero.
  - One assignment may be exempted only if a University health services certificate is given to the professor within one week of the due date.

- Absence from midterm / final examination: see Engineering faculty regulations.
  - Illness requires University health services certificate
  - Travel, employment, misreading the timetable are not valid excuses

Academic Integrity

- What is academic fraud?
  - Misrepresenting someone else's work as your own:
    - Failure to cite sources, including the internet and discussions.
    - Use of the words of someone else without quotation marks or other highlighting.
  - Falsified lab data or citations.
  - Violation of examination regulations.
  - Tampering with academic evaluations.
  - Helping another student do any of the above.

Solving problems with computing



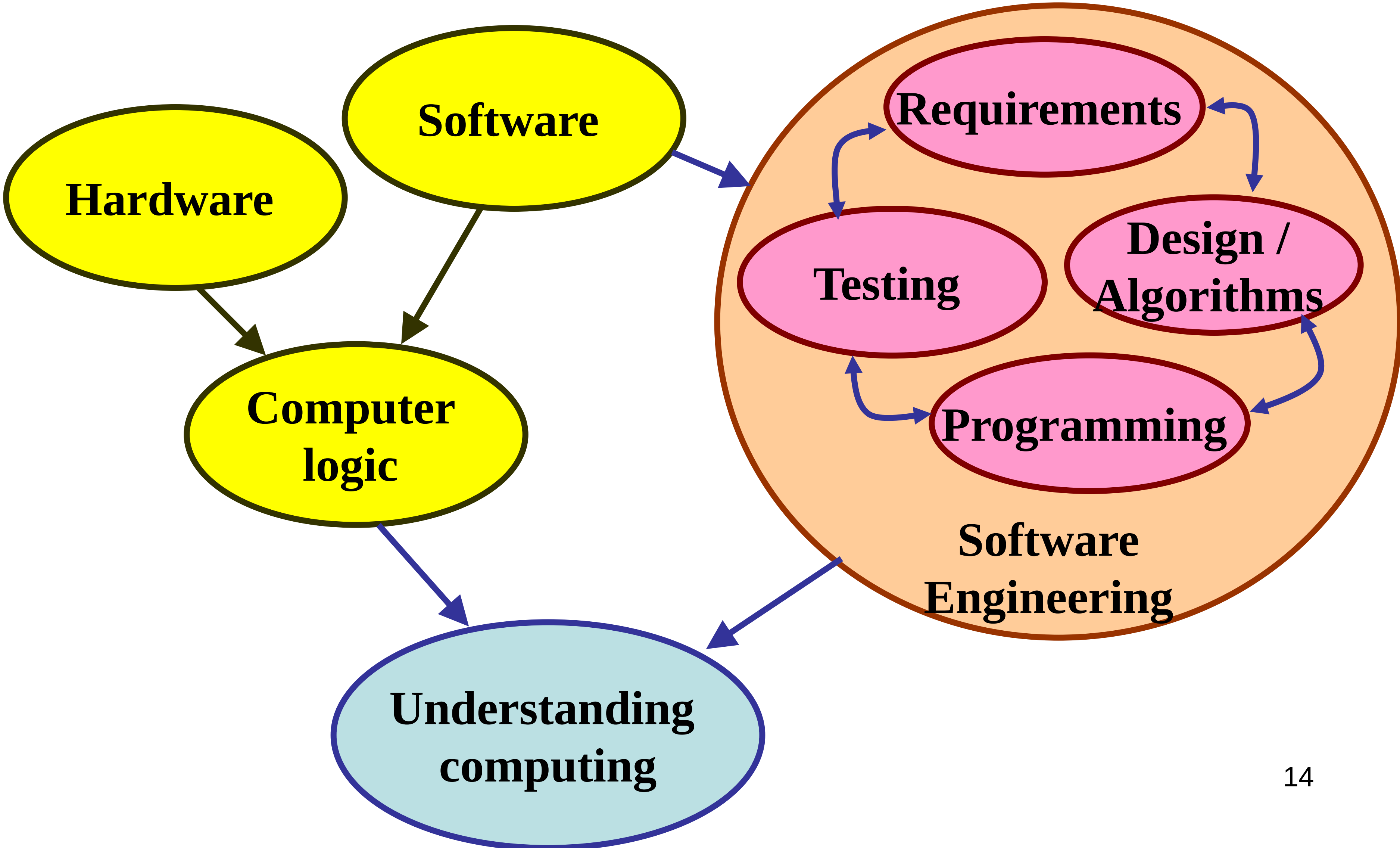
Understanding the problem

Understanding the problem domain

Understanding computing

A solution

Some Computing Concepts

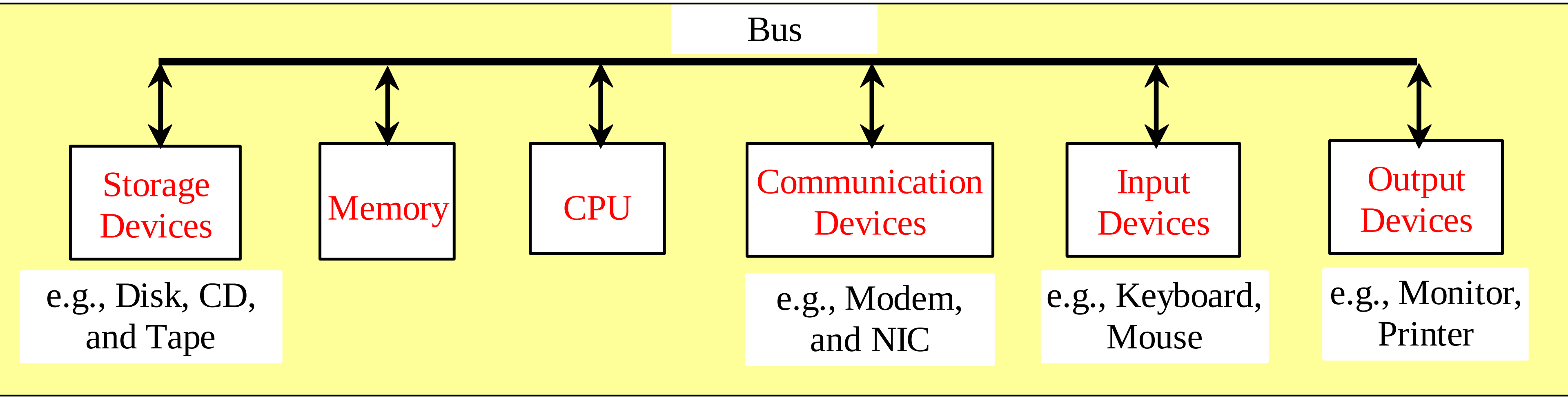


What is Computer Logic?

- A computer processes 0’s and 1’s
- How is the computer organized to provide a tool for computation and solving problems?

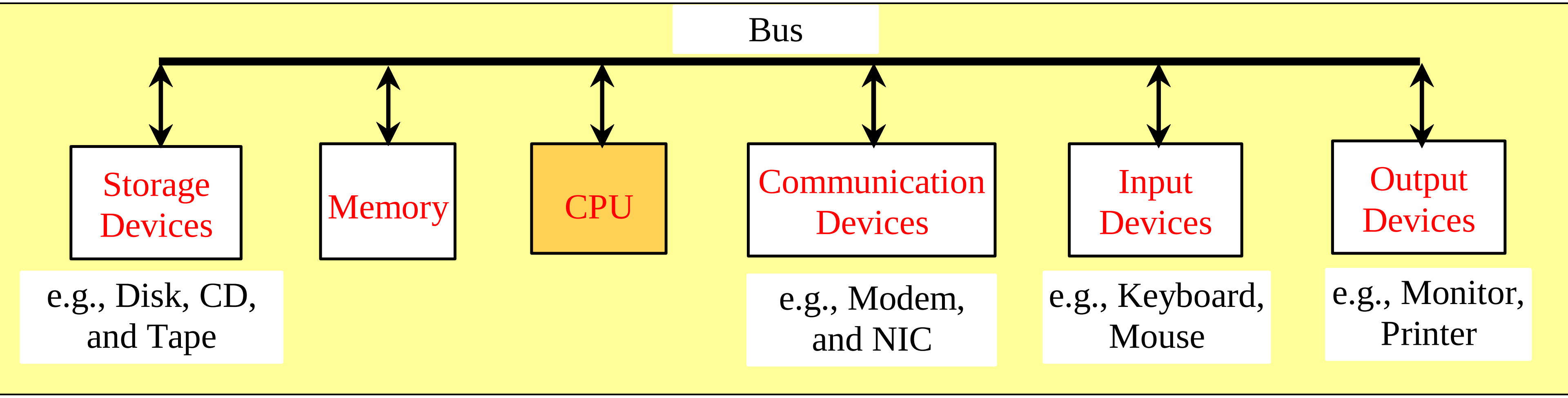
What is a Computer?

A computer consists of a CPU, memory, hard disk, floppy disk, monitor, printer, and communication devices.



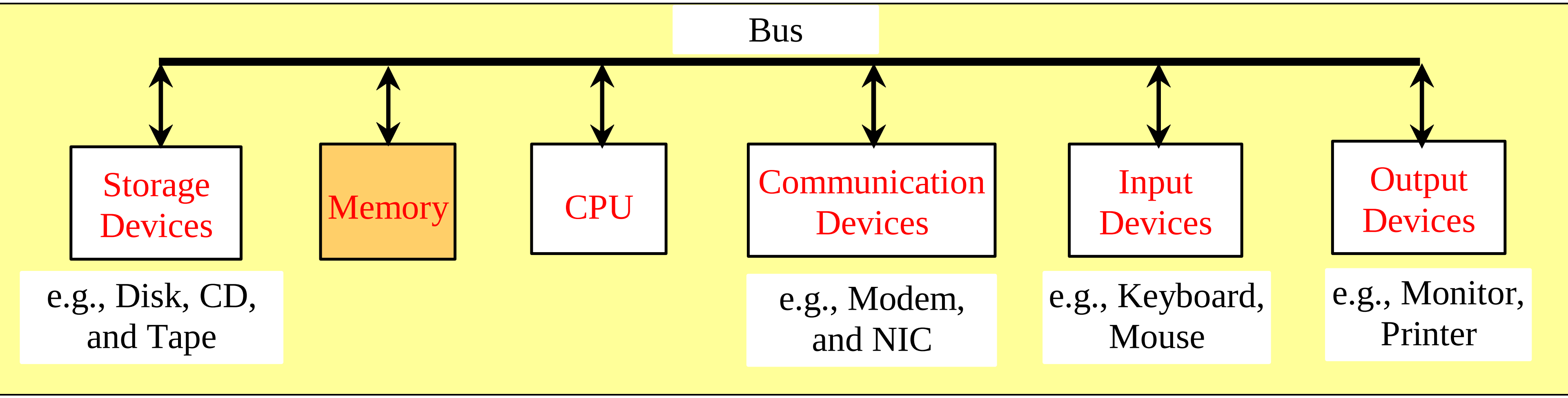
CPU

The central processing unit (CPU) is the brain of a computer. It retrieves instructions from memory and executes them. The CPU speed is measured in megahertz (MHz), with 1 megahertz equaling 1 million pulses per second. The speed of the CPU has been improved continuously. If you buy a PC now, you can get an Intel Pentium 4 Processor at 3 gigahertz (1 gigahertz is 1000 megahertz).



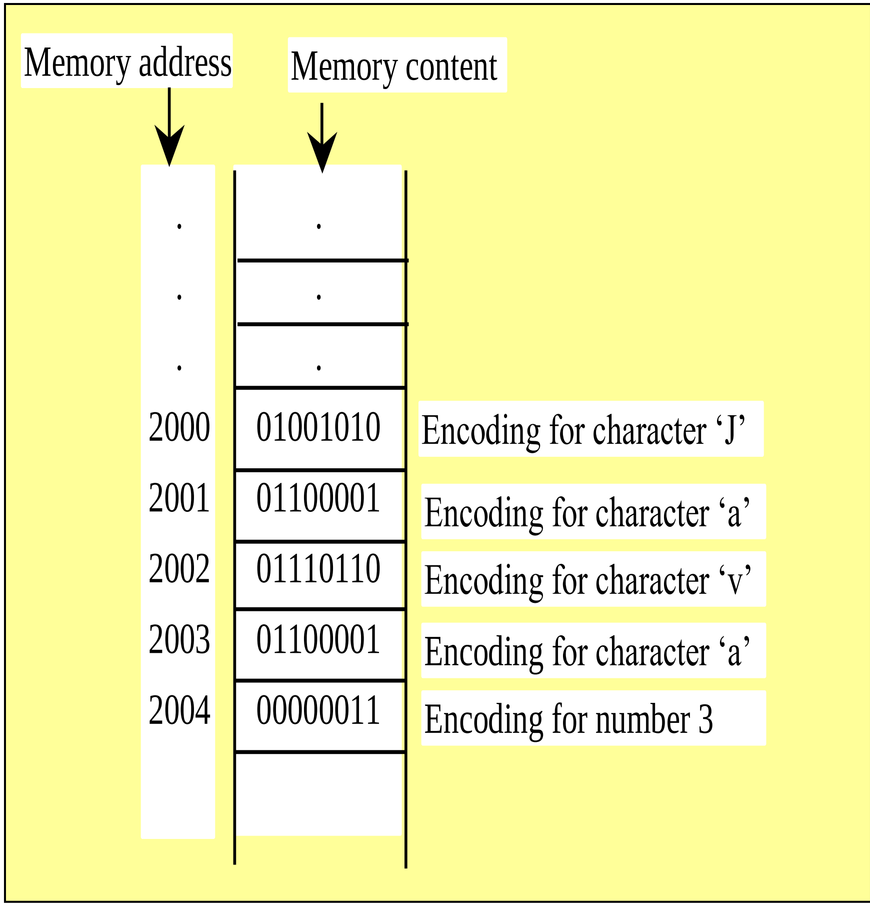
Memory

Memory is to store data and program instructions for CPU to execute. A memory unit is an ordered sequence of bytes, each holds eight bits. A program and its data must be brought to memory before they can be executed. A memory byte is never empty, but its initial content may be meaningless to your program. The current content of a memory byte is lost whenever new information is placed in it.



How Data is Stored?

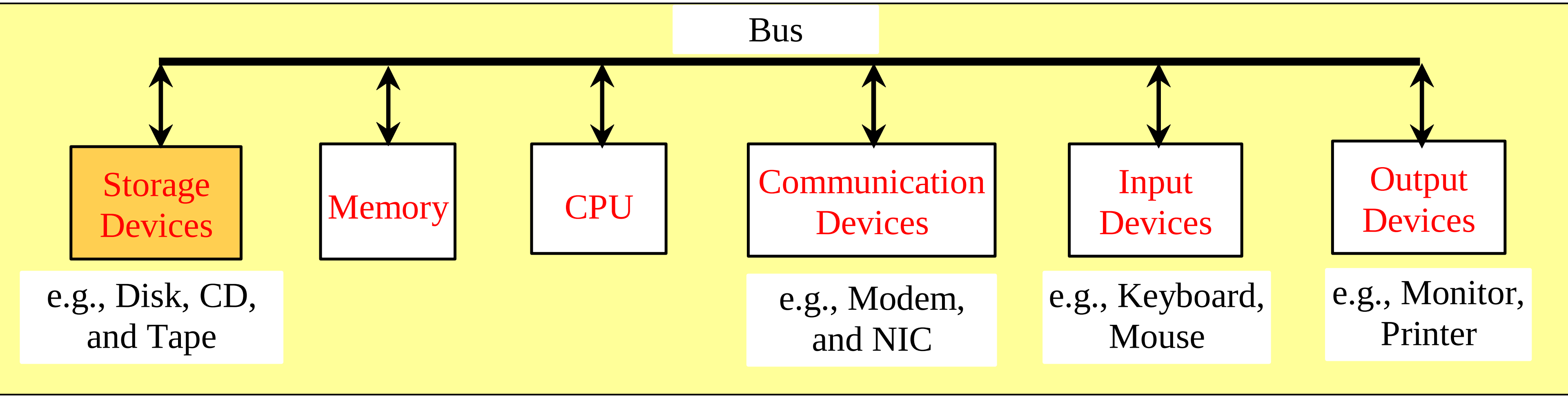
Data of various kinds, such as numbers, characters, and strings, are encoded as a series of bits (zeros and ones). Computers use zeros and ones because digital devices have two stable states, which are referred to as *zero* and *one* by convention. The programmers need not to be concerned about the encoding and decoding of data, which is performed automatically by the system based on the encoding scheme. The encoding scheme varies. For example, character 'J' is represented by 01001010 in one byte. A small number such as three can be stored in a single byte. If computer needs to store a large number that cannot fit into a single byte, it uses a number of adjacent bytes. No two data can share or split a same byte. A byte is the minimum storage unit.



Memory address	Memory content	
.	.	
.	.	
.	.	
2000	01001010	Encoding for character 'J'
2001	01100001	Encoding for character 'a'
2002	01110110	Encoding for character 'v'
2003	01100001	Encoding for character 'a'
2004	00000011	Encoding for number 3

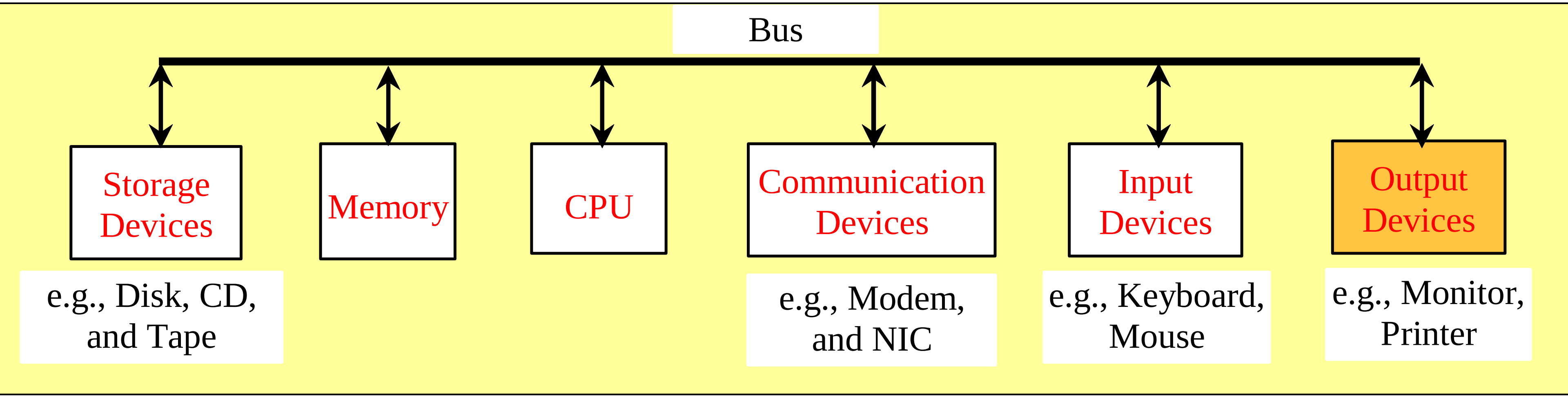
Storage Devices

Memory is volatile, because information is lost when the power is off. Programs and data are permanently stored on storage devices and are moved to memory when the computer actually uses them. There are three main types of storage devices:Disk drives (hard disks and floppy disks), CD drives (CD-R and CD-RW), and Tape drives.



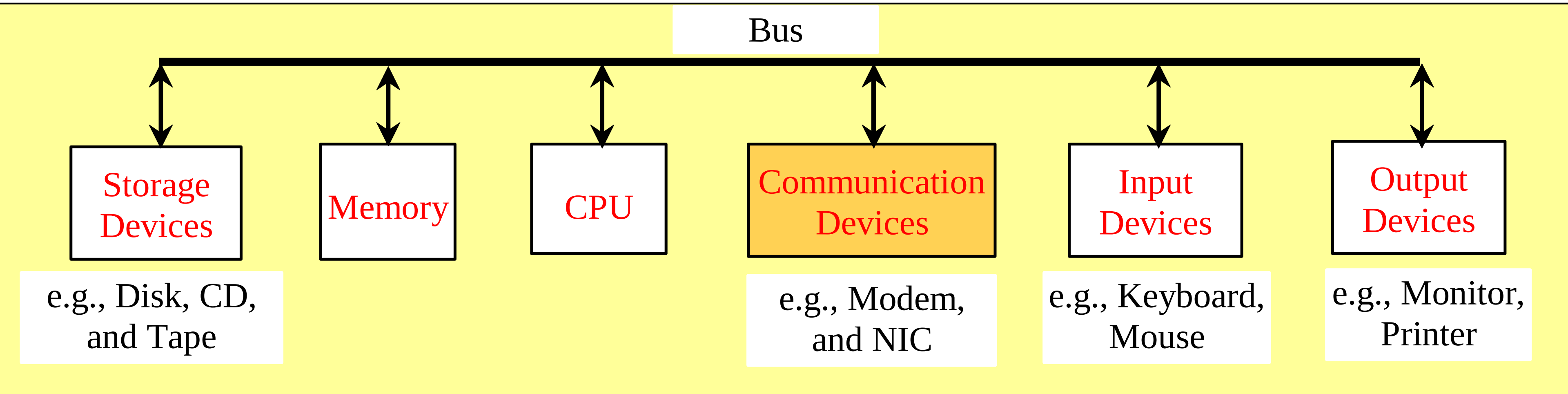
Output Devices: Monitor

The monitor displays information (text and graphics). The resolution and dot pitch determine the quality of the display.



Communication Devices

A *regular modem* uses a phone line and can transfer data in a speed up to 56,000 bps (bits per second). A *DSL* (digital subscriber line) also uses a phone line and can transfer data in a speed 20 times faster than a regular modem. A *cable modem* uses the TV cable line maintained by the cable company. A cable modem is as fast as a DSL. Network interface card (*NIC*) is a device to connect a computer to a local area network (LAN). The LAN is commonly used in business, universities, and government organizations. A typical type of NIC, called *10BaseT*, can transfer data at 10 mbps (million bits per second).



Programs

Computer *programs*, known as *software*, are instructions to the computer.

You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

Programs are written using programming languages.

Programming Languages

Machine Language   Assembly Language   High-Level Language

Machine language is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Program with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify. For example, to add two numbers, you might write an instruction in binary like this:

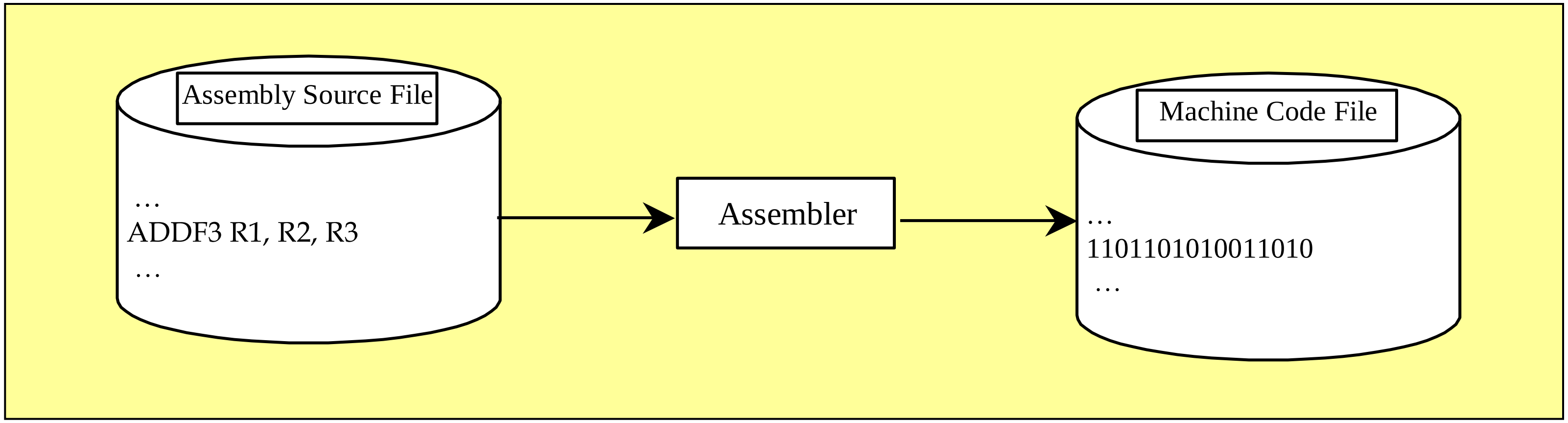
```
1101101010011010
```

Programming Languages

Machine Language Assembly Language High-Level Language

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code. For example, to add two numbers, you might write an instruction in assembly code like this:

ADDF3 R1, R2, R3



Programming Languages

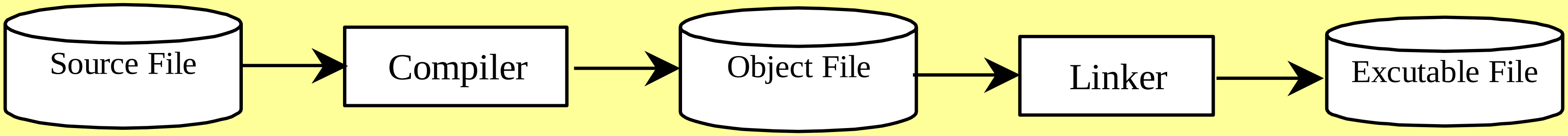
Machine Language   Assembly Language   High-Level Language

The high-level languages are English-like and easy to learn and program. For example, the following is a high-level language statement that computes the area of a circle with radius 5:

```
        area = 5 * 5 * 3.1415;
```

Compiling Source Code

A program written in a high-level language is called a *source program*. Since a computer cannot understand a source program. Program called a *compiler* is used to translate the source program into a machine language program called an *object program*. The object program is often then linked with other supporting library code before the object can be executed on the machine.

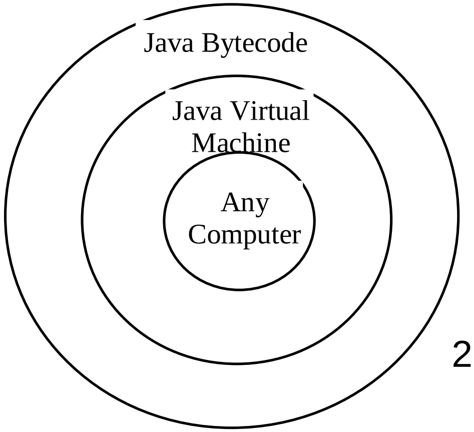


Popular High-Level Languages

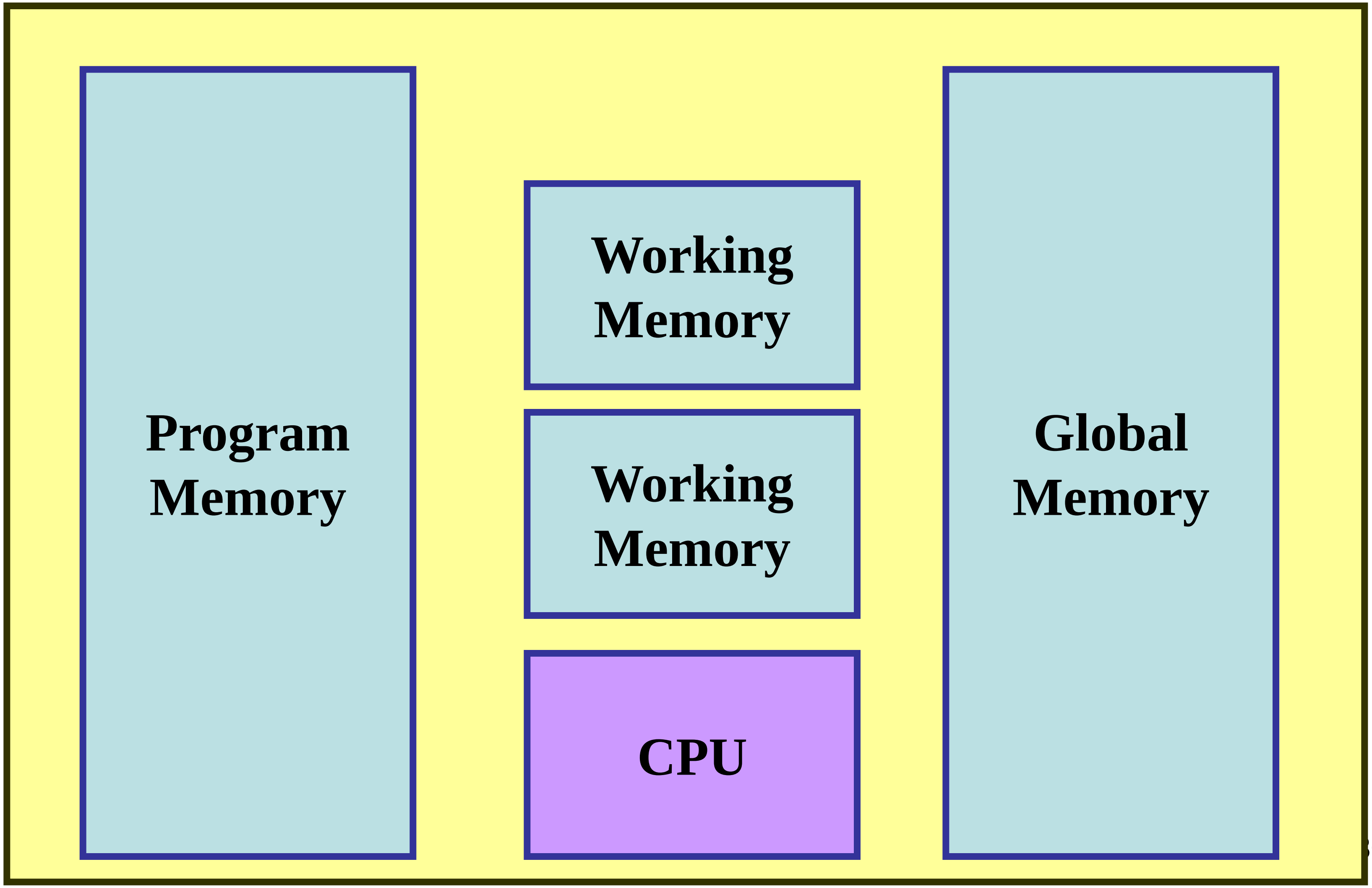
- COBOL (COmmon Business Oriented Language)
- FORTRAN (FORmula TRANslation)
- BASIC (Beginner All-purpose Symbolic Instructional Code)
- Pascal (named for Blaise Pascal)
- Ada (named for Ada Lovelace)
- C (whose developer designed B first)
- Visual Basic (Basic-like visual language developed by Microsoft)
- Delphi (Pascal-like visual language developed by Borland)
- C++ (an object-oriented language, based on C)
- Java (We use it in this course)

Compiling Java Source Code

You can port a source program to any machine with appropriate compilers. The source program must be recompiled, however, because the object program can only run on a specific machine. Nowadays computers are networked to work together. Java was designed to run object programs on any platform. With Java, you write the program once, and compile the source program into a special type of object code, known as *bytecode*. The bytecode can then run on any computer with a Java Virtual Machine, as shown in the Figure, Java Virtual Machine is a software that interprets Java bytecode.

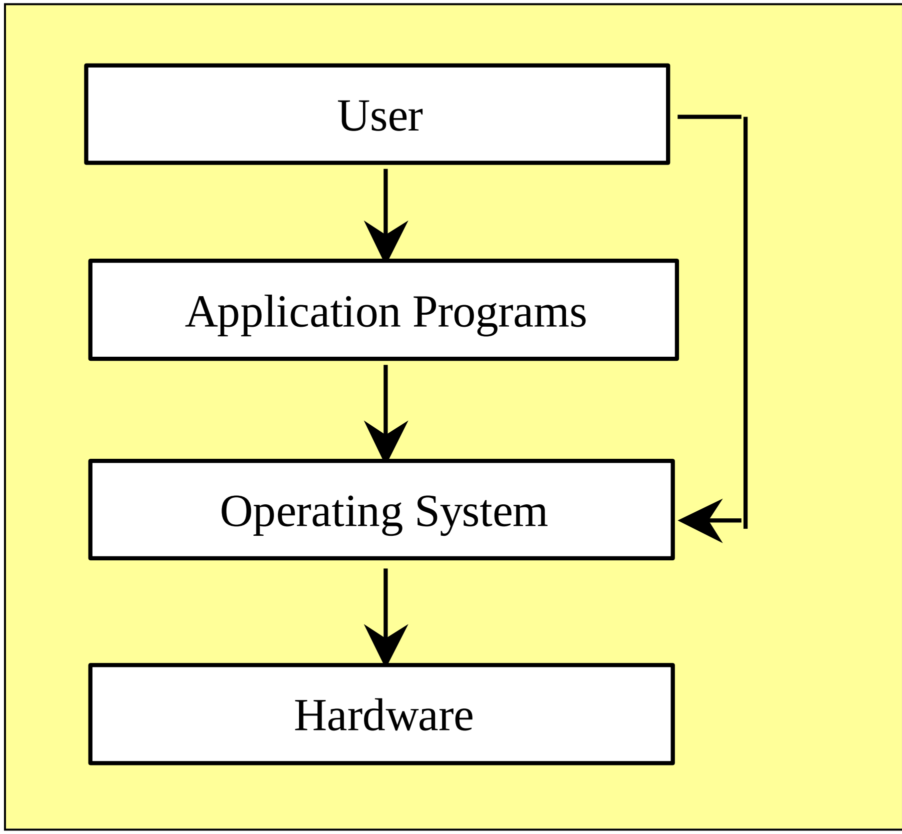


Programming Model



Operating Systems

The *operating system* (OS) is a program that manages and controls a computer's activities. You are probably using Windows 98, NT, 2000, XP, or ME. Windows is currently the most popular PC operating system. Application programs such as an Internet browser and a word processor cannot run without an operating system.


User
Application Programs
Operating System
Hardware