Computational Thinking and Programing

Introduction to Computer Science, Algorithm, H/W and S/W

Contents

- Introduction to Computer Science
- Algorithm
- Hardware
- Software
- Programming Languages

What is Computer Science

- *Computer science is fundamentally about is computational problem solving.*
  - Solving problems by the use of computation

- **Areas of study in computer science includes:**
  - Software engineering, database management, computer networks, data mining, information security, programming language design, computer architecture, human–computer interaction, robotics, and artificial intelligence, among others.

- **What is computation?**
- ***Characterization of computation can be given by the notion of an algorithm.***
- ***Algorithm:*** Series of steps that can be systematically followed for producing the answer to a certain type of problem.

Computational Problem Solving

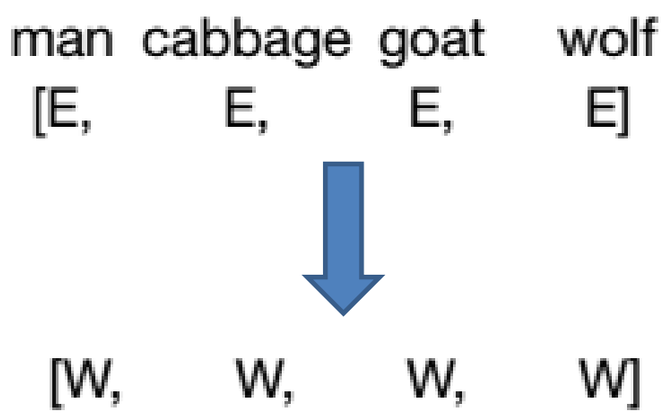
- *Two things are needed to solve a problem computationally:*
  - ***Representation:*** *captures all the relevant aspects of the problem,*
  - ***Algorithm:*** *solves the problem by use of the representation*

- **Example: Man, Cabbage, Goat, Wolf problem**
  - A man lives on the east side of a river. He wishes to bring a cabbage, a goat, and a wolf to a village on the west side of the river to sell.
- **Constraints:**
  - Boat can only hold himself, and either the cabbage, goat, or wolf.
  - Man cannot leave the goat/ wolf alone with Cabbage/goat, as the goat/wolf will eat the Cabbage /goat.

Man, Cabbage, Goat, Wolf problem: Solution

- *Simple algorithmic approach for solving this problem is trying all possible combinations of items that may be rowed back (**called brute force**).*
- **Representation for this problem:**
  - Relevant aspects of problem need to be represented, all irrelevant details can be omitted.
  - Relevant information is ***where*** *each item is at each step,* ***or*** *what is* ***state*** *of each item at each step*.

- The *start state* of the problem can be represented as follows:

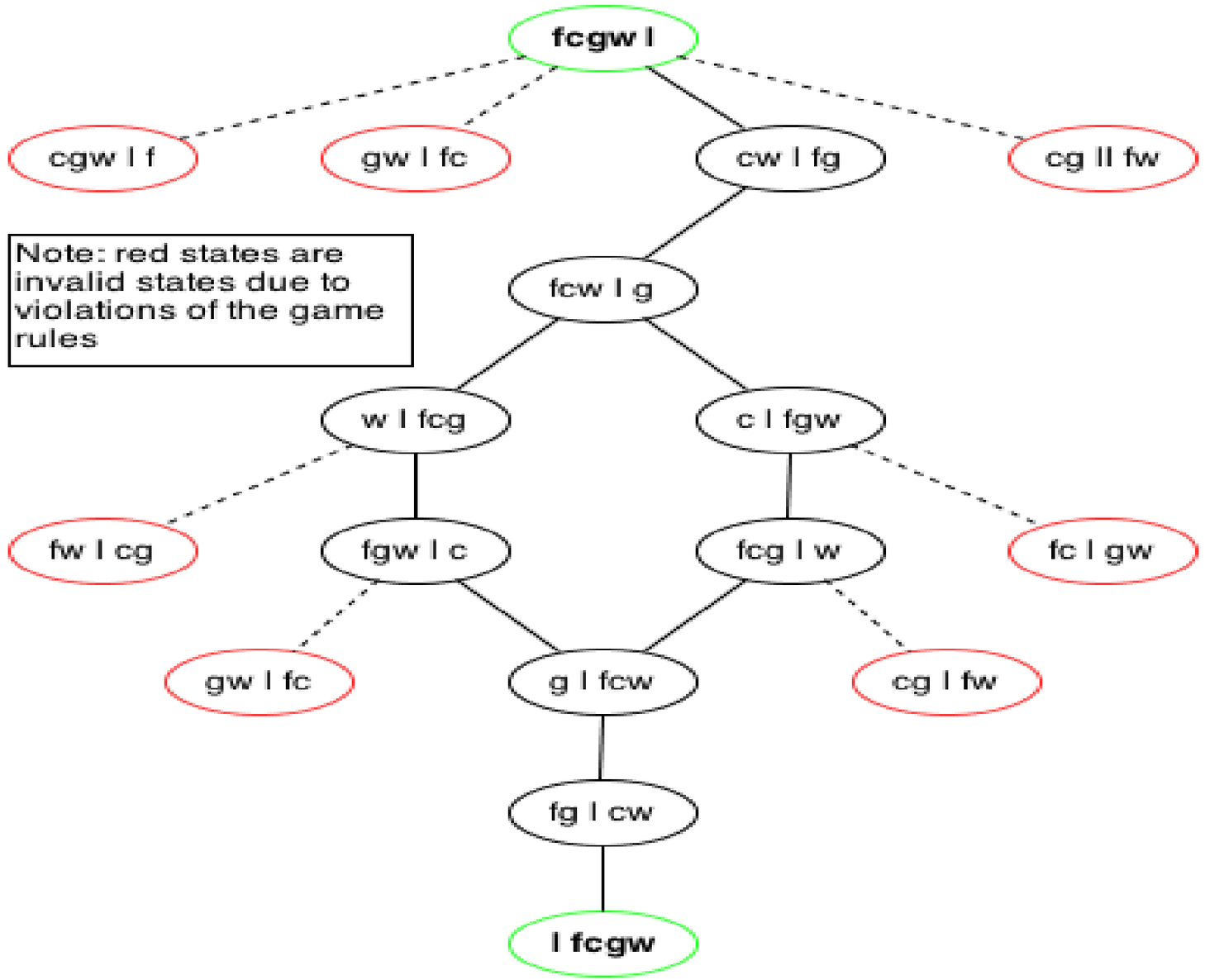


Man, Cabbage, Goat, Wolf problem: Solution

man cabbage goat wolf

[E, E, E, E]

[W, W, W, W]



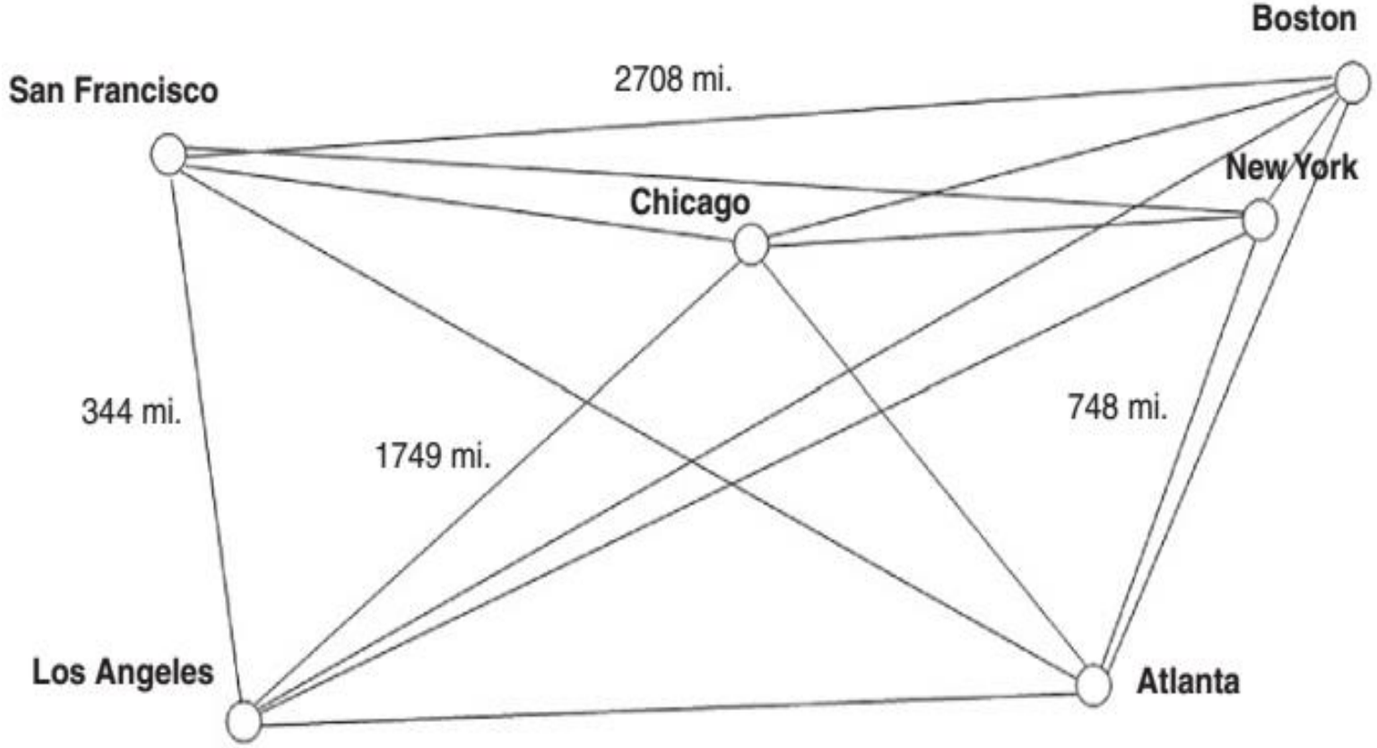
Limits of Computational Problem Solving

- After the development of an algorithm for solving a given problem, an important question is,

  "Can a solution to the problem be found in a reasonable amount of time?"

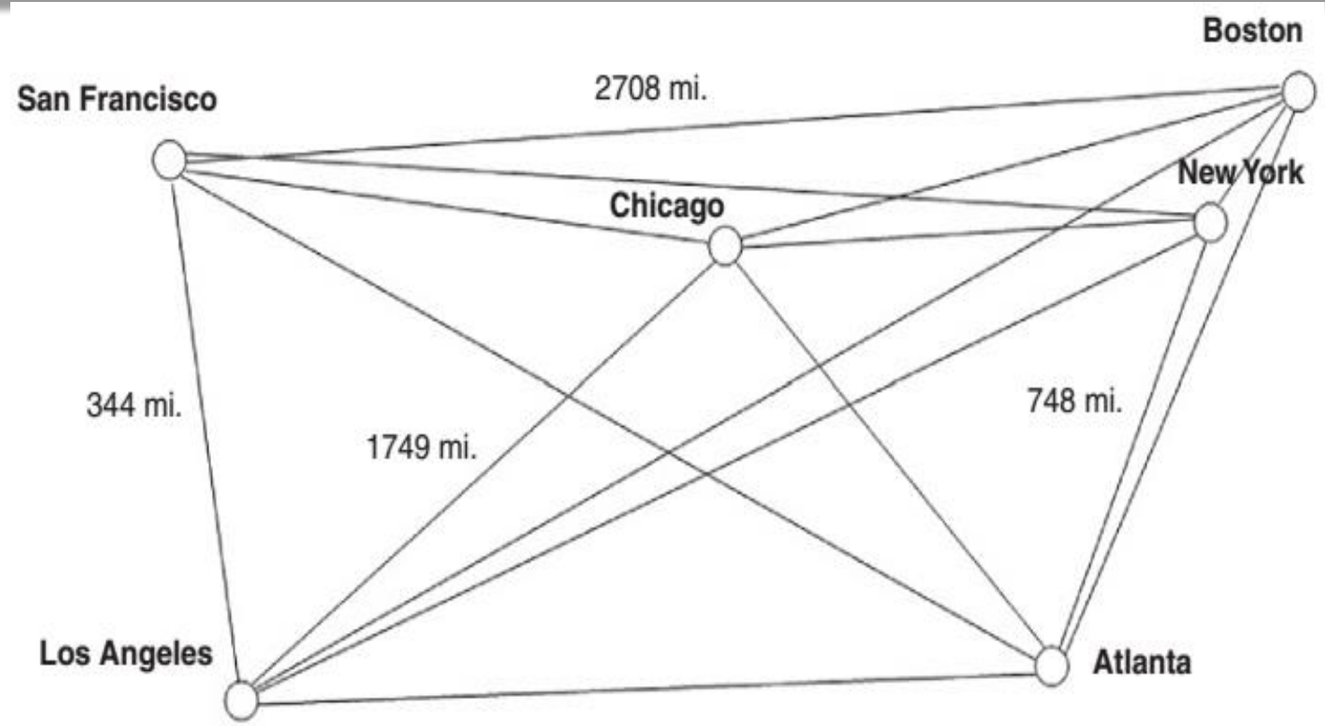
- *If not, then the particular algorithm is of limited practical use.*

- ***Example:* Traveling Salesman problem**
  - The problem is to find the shortest Route of travel for a salesman needing to visit a given set of cities.



Limits of Computational Problem Solving

- ***Example:*** **Traveling Salesman problem**
- **Brute force approach:** The lengths of all possible routes would be calculated and compared to find the shortest one.



- For 10 cities, the number of possible routes is 10!, whereas for 20 cities, the number of possible routes is 20!.
- If a computer could compute the lengths of one million routes per second, it would take over 77,000 years to find the shortest route for twenty cities by this approach. Therefore, brute-force approach is impractical for this problem.

- ***Any algorithm that correctly solves a given problem must solve the problem in a reasonable amount of time, otherwise it is of limited practical use.***

What is an Algorithm ?

- *An algorithm is a **finite** number of clearly described, unambiguous "feasible" steps that can be systematically followed to produce a desired result for given input in a **finite** amount of time. (that is, it eventually terminates).*

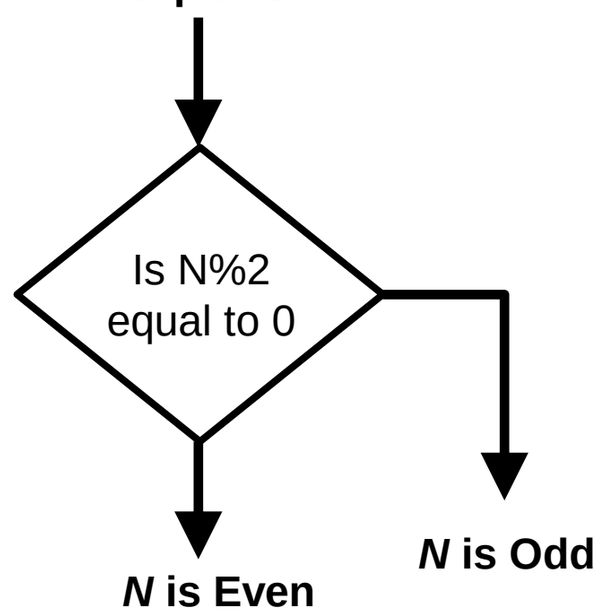
- Computer algorithms are central to computer science. They provide step-by-step methods of computation that a machine can carry out.
- **Example: Write an algorithm to check a given number is even or odd.**

1. Let ***N*** be the number to check whether it is even or odd.

2. Divides the ***N*** by **2** and check the remainder

3. If ***N*** divided by **2** provides 0 remainder, then N is **even**,

4. Else **N** is **odd.**



Computer Algorithms

- The computation that a given computer performs is only as good as the underlying algorithm used.

- Since computers can execute instructions very quickly and reliably without error, algorithms and computers are a perfect match.

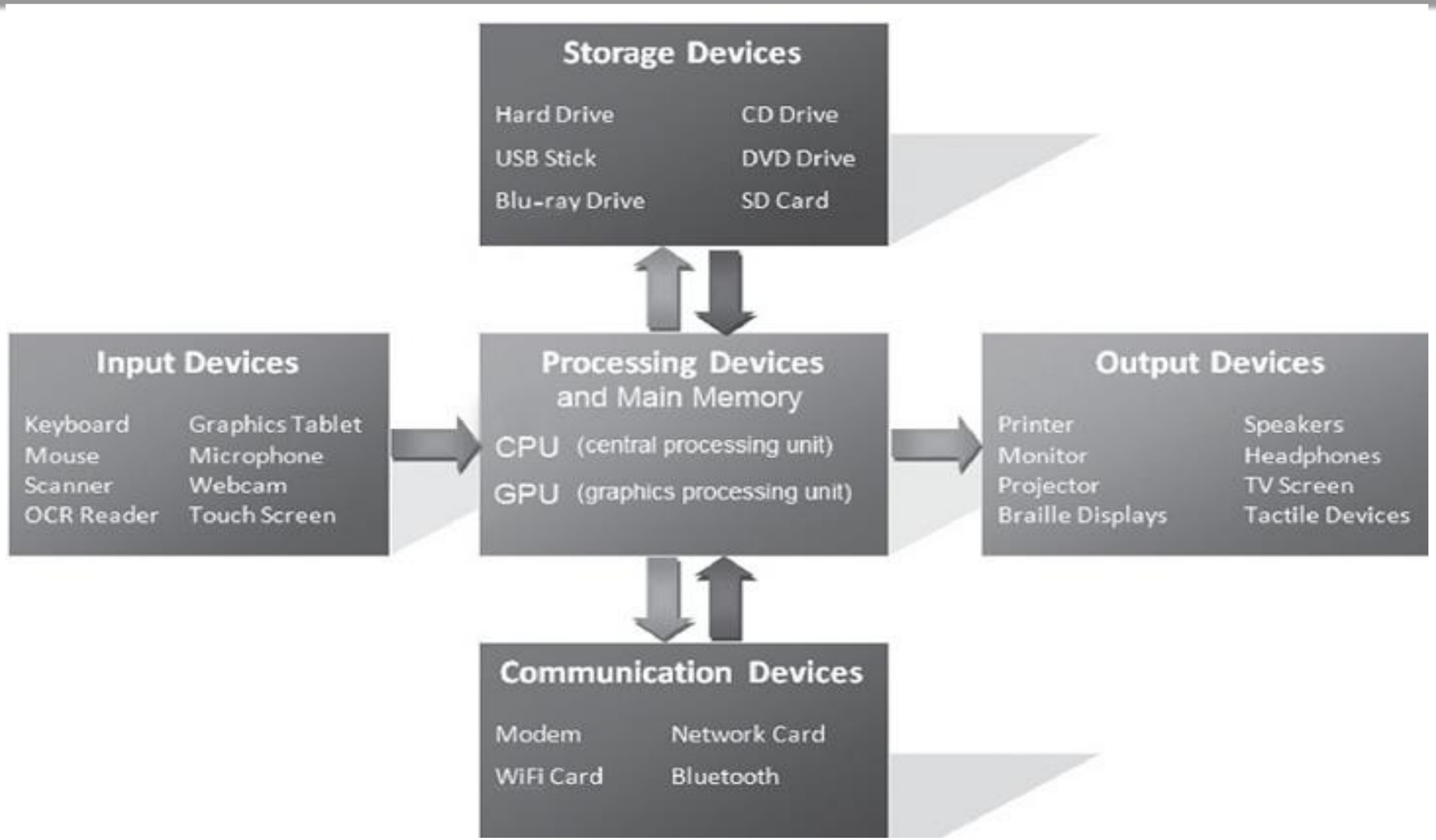
- **Exercises:**
  1. **Write an Algorithm to check whether a given number is prime**

  2. **Write an algorithm for determining the day of the week for any date between January 1, 1800 and December 31, 2099.**

Computer Hardware

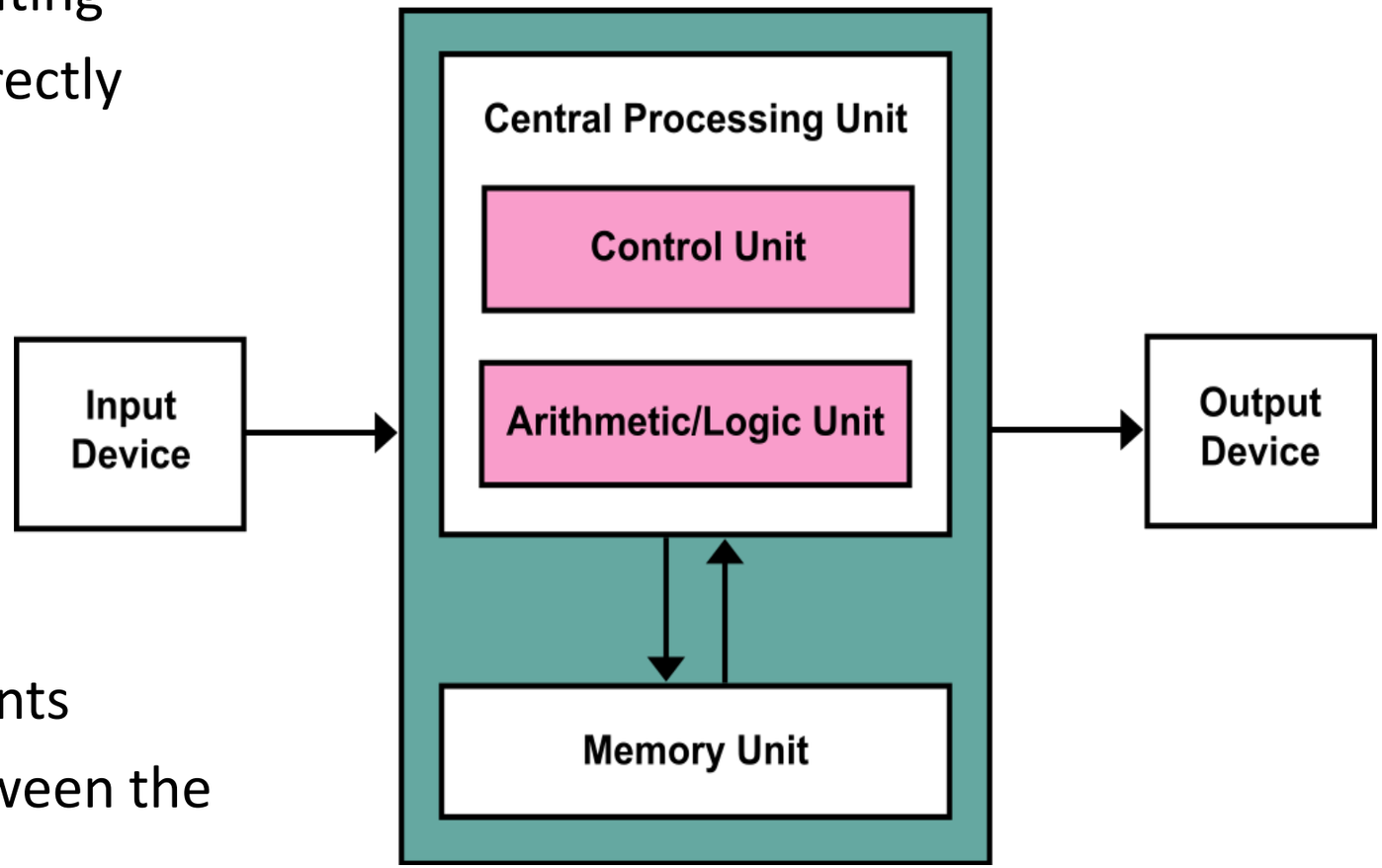
- *Computer hardware comprises the physical part of a computer system. It includes the all-important components of the central processing unit (CPU) and main memory.*
- *It basically includes following peripheral components:*
  - **Input devices** − keyboard, mouse, etc.
  - **Output devices** − printer, monitor, etc.
  - **Secondary storage devices** − Hard disk, CD, DVD, etc.
  - **Internal components** − CPU, motherboard, RAM, etc.

Fundamental Hardware Components



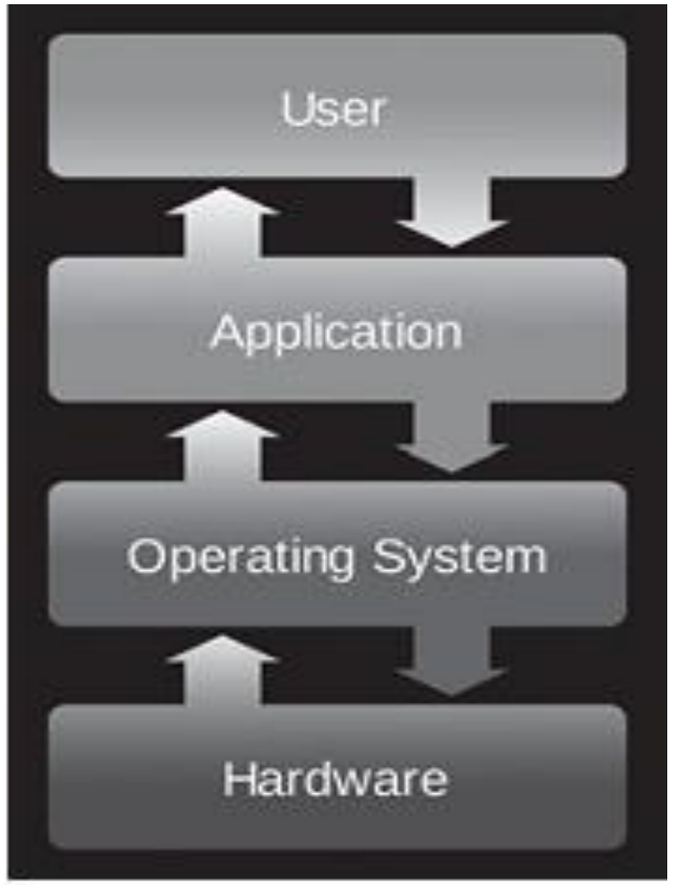
Bridging Hardware and Software

- *The **central processing unit** (CPU) is the "brain" of a computer, containing digital logic circuitry able to interpret and execute instructions.*
- **Main** memory is where currently executing programs reside, which the CPU can directly and very quickly access.
- **Secondary** memory is nonvolatile, and therefore provides long-term storage of programs and data
- **Input/output** devices include anything that allows for input or output.
- **Buses** transfer data between components within a computer system, such as between the CPU and main memory.



Bridging Hardware and Software

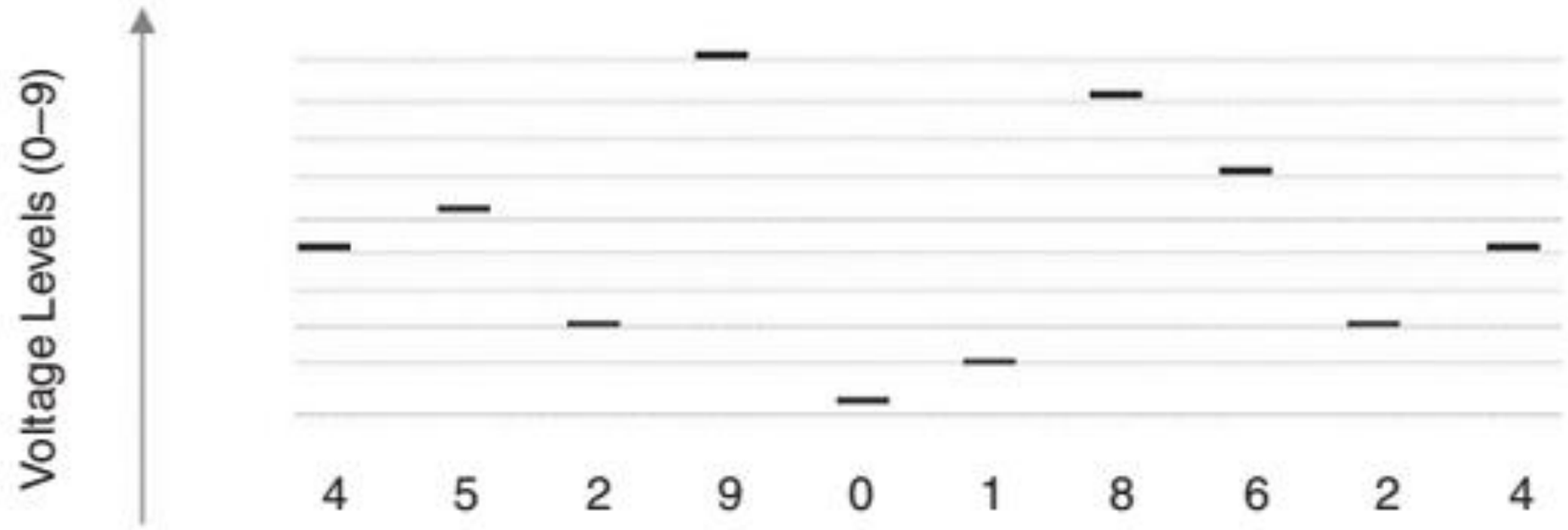
- Hardware and software are mutually dependent on each other. Both of them must work together to make a computer produce a useful output.
- An **operating system** is software that has the job of managing the hardware resources of a given computer and providing a particular user interface (interacting with the hardware resources).
- An operating system is intrinsic to the operation a computer, it is referred to as **system software**.
- Operating system acts as the “middle man” between the hardware and executing application programs



- **Limits of Integrated Circuits Technology:**
  Moore’s Law states that the number of transistors that can be placed on a single silicon chip doubles roughly every two years.

Digital Computing: Information Representation

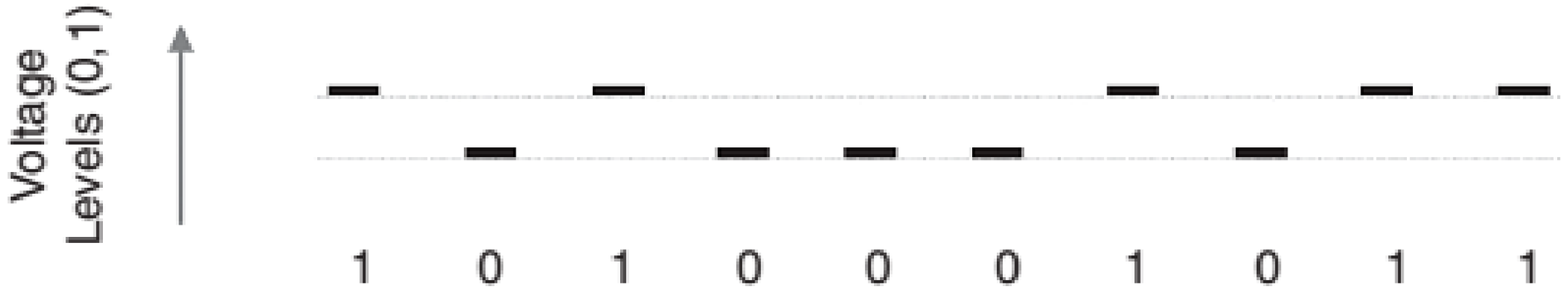
- In digital computing, all information is represented as a series of digits.
- We are used to representing numbers using base 10 with digits 0–9. Since in current electronic computing, each digit is represented by a different voltage level, the information can be represented within a computer system as given follows



- The more voltage levels (digits) that the hardware must utilize and distinguish, the more complex the hardware design becomes.

Information Representation

- It is a fact of information theory, that any information can be represented using only two symbols.
- Therefore, all information within a computer system is represented using only two digits, 0 and 1, called **binary representation**.



- Computer hardware, therefore, is based on the use of simple electronic “on/off” switches called **transistors** that switch at very high speed.
- **Integrated circuits** (“chips”), the building blocks of computer hardware, are comprised of millions or even billions of transistors.

Binary Number System

- For representing numbers, any base (radix) can be used. In base 10, there are ten possible digits (0, 1, . . ., 9), in which each column value is a power of ten, as shown below.

10,000,000	1,000,000	100,000	10,000	1,000	100	10	1
10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0
						9	**9** = **99**

- Other radix systems work in a similar manner. Base 2has digits 0 and 1, with place values that are powers of two, as given below.

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	**1**	**1**	**0**	**0**	**0**	**1**	**1**
0 +	64 +	32 +	0 +	0 +	0 +	2 +	1 = **99**

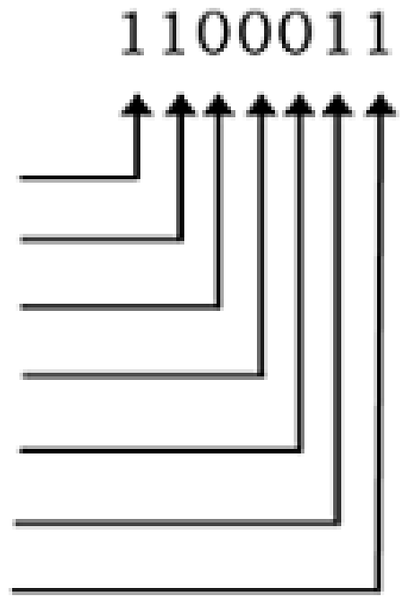
- The term **bit** stands for binary digit. Therefore, every bit has the value 0 or 1. A **byte** is a group of bits operated on as a single unit in a computer system, usually consisting of eight bits.

Conversion Decimal (base 10) to Binary (base 2)

- The algorithm for the conversion from base 10 to base 2 is to successively divide a number by two until the remainder becomes 0.

- The remainder of each division provides the next higher-order (binary) digit

```
99/2   =   49, with remainder 1
49/2   =   24, with remainder 1
24/2   =   12, with remainder 0
12/2   =   6,  with remainder 0
6/2    =   3,  with remainder 0
3/2    =   1,  with remainder 1
1/2    =   0,  with remainder 1
```



- The binary representation of 99 to be 1100011.

What Is Computer Software?

- ***Computer software*** *is a set of program instructions, including related data and documentation, that can be executed by computer.*
- ***Software*** *can be in the form of instructions on paper, or in digital form.*

- ***Types of Software:***
  - ***System Software:*** *intrinsic to a computer system or platform for executing application software. Example: OS, Compiler, Interpreter, etc.*
  - ***Application software:*** *Fulfills users' needs, such as a photo-editing program, MS office, etc.*

- Programming languages (called "artificial languages") are languages just as "natural languages" such as English and Mandarin (Chinese).
- Syntax and semantics are important concepts that apply to all languages.

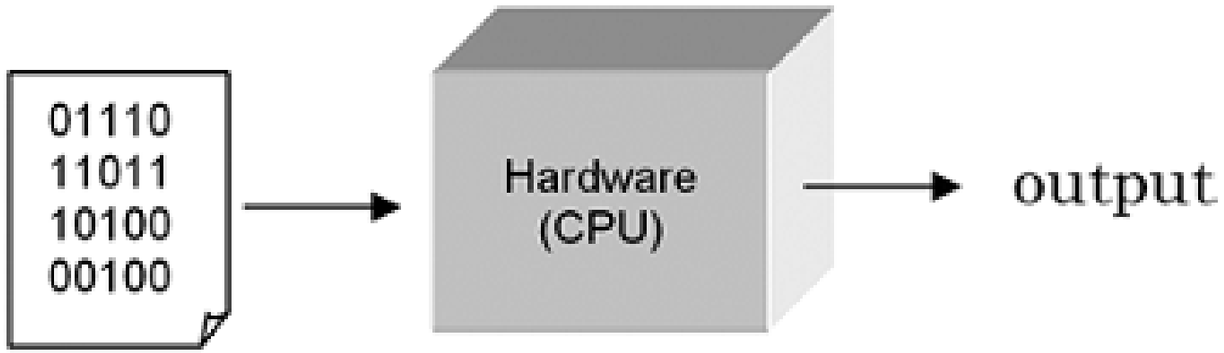
Syntax, Semantics of a Language

- The **syntax** of a language is a set of characters and the acceptable sequences of those characters.
- The **semantics** of a language is the meaning associated with each syntactically correct sequence of characters.
- **Example:**
  - “Hello there, how are you?” -- This is not syntactically correct,
  - “Hello there, hao are you?” -- “hao” is not a word in the English language
  - “Colorless green ideas sleep furiously.” -- Syntactically correct, but semantically incorrect (no meaning).

ENGLISH	MANDARIN (pinyin)	MANDARIN (Chinese Characters)
Syntax Hao	**Syntax** Hao	**Syntax** 好
Semantics No meaning *(syntactically incorrect)*	**Semantics** *“Good”*	**Semantics** *“Good”*

Program Translation

- A central processing unit (CPU) is designed to interpret and execute a specific set of instructions represented in binary form (i.e., 1s and 0s) called **machine code**.
- Only programs in machine code can be executed by a CPU.

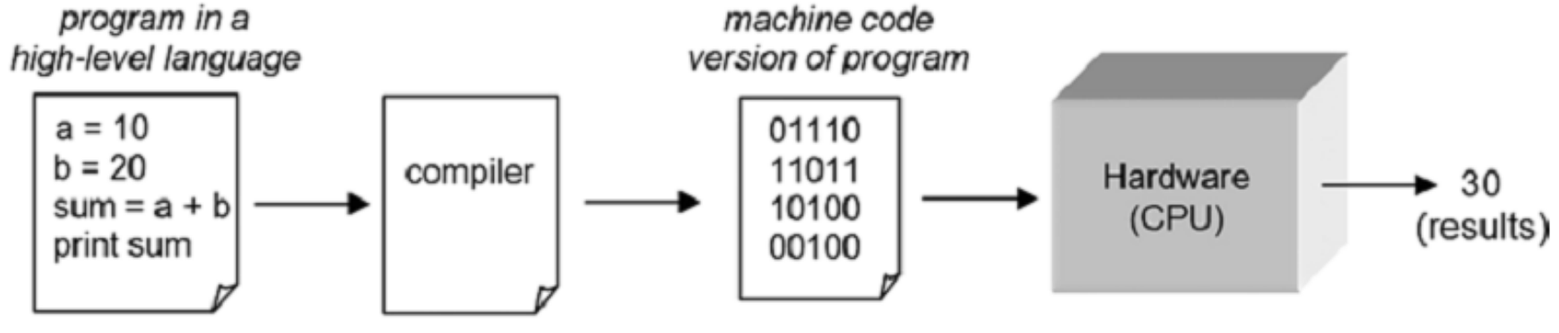


- Writing programs at this “low level” is tedious and error-prone. Therefore, most programs are written in a “high-level” programming language such as Python.

- Since the instructions of such programs are not in machine code that a CPU can execute, a translator program must be used.
- There are two fundamental types of translators: **compiler** and **interpreter**.

Program Translation

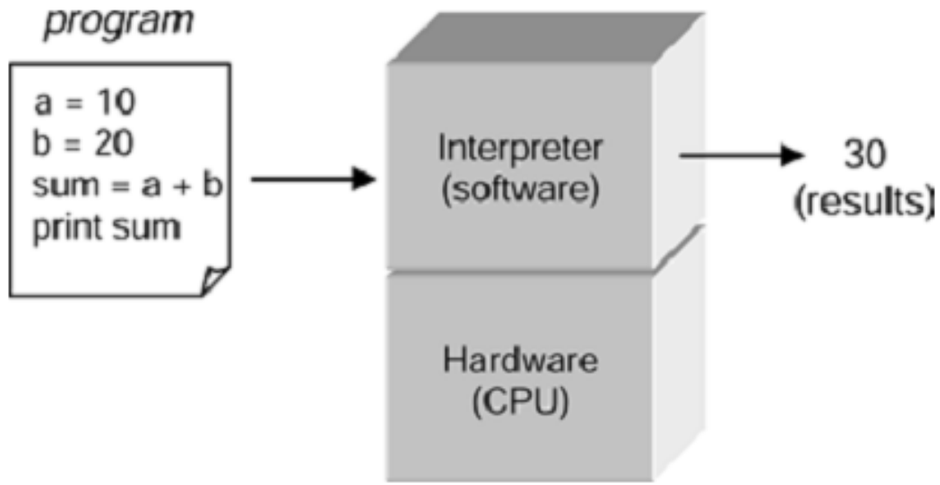
- A compiler is a translator program that translates programs directly into machine code to be executed by the CPU.



- An interpreter executes program instructions in place of (“running on top of”) the CPU. It can immediately execute instructions as they are entered.

- Python, as we shall see, is executed by an interpreter.

- Compiled programs generally execute faster than interpreted programs.



Program Debugging: Syntax Errors vs. Semantic Errors

- **Program debugging** is the process of finding and correcting errors **("bugs")** in a computer program.
- **Syntax errors** are caused by invalid syntax (for **E.g.,** entering prnt instead of print)
- Since a translator cannot understand instructions containing syntax errors, translators terminate when encountering such errors indicating where the problem occurred in program.
- **Semantic errors** (generally called logic errors) are caused by errors in program logic. These errors cannot be automatically detected, as translators cannot understand the intent of a given computation.
- **Example:** if a program computed the average of three numbers as follows,

  (num1 + 1num2 + 1num3) / 2.0

- A translator would have no means of determining that the divisor should be 3 and not 2.

- ***Computers do not understand what a program is meant to do, they only follow the instructions given.***

Procedural vs. Object-Oriented Programming

- **Procedural programming** and **object-oriented programming** are two major programming paradigms in use today.
- Each provides a different way of thinking about computation.
- While most programming languages only support one paradigm.
- Python supports both procedural and object-oriented programming

Exercise: MCQs

1. Two general types of software are system software and _____ software.
2. The syntax of a given language is,
   a) the set of symbols in the language.
   b) the acceptable arrangement of symbols.
   c) both of the above
3. The semantics of a given language is the meaning associated with any arrangement of symbols in the language.
   a) TRUE
   b) FALSE
4. CPUs can only execute instructions that are in binary form called _____.
5. The two types of translation programs for the execution of computer programs are _____ and _____.
6. The process of finding and correcting errors in a computer program is called _____.
7. Which kinds of errors can a translator program detect?
   a) Syntax errors
   b) Semantic errors
   c) Neither of the above
8. Two major programming paradigms in use today are _____ programming and _____ programming.

MCQs: Answers

1. Two general types of software are system software and **application** software.
2. The syntax of a given language is,
   a) the set of symbols in the language.
   b) the acceptable arrangement of symbols.
   c) both of the above
3. The semantics of a given language is the meaning associated with any arrangement of symbols in the language.
   a) TRUE
   b) FALSE
4. CPUs can only execute instructions that are in binary form called **machine code**.
5. The two types of translation programs for the execution of computer programs are **compilers** and **interpreters**.
5. The process of finding and correcting errors in a computer program is called **program debugging**.
6. Which kinds of errors can a translator program detect?
   a) Syntax errors
   b) Semantic errors
   c) Neither of the above
7. Two major programming paradigms in use today are **procedural** programming and **object-oriented** programming.

Thank You