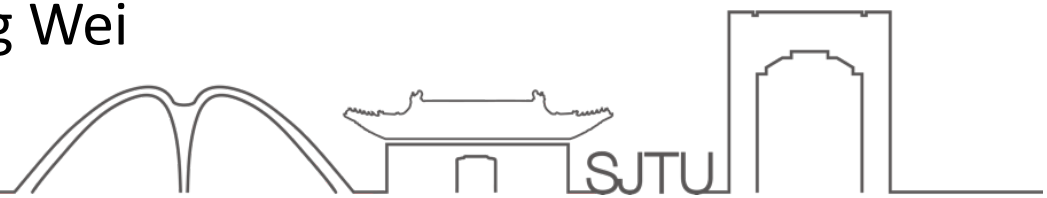
JOINT INSTITUTE
交大密西根学院

VG101/ENGR1010J S2: Introduction to Computers and Programming

Lecture 01

Xiaodong Wei



Outline

- Course information
- Introduction to computers and programming

Course Information

Using Canvas

- https://jicanvas.com
- Class materials are managed by Canvas
- Important announcements, such as time and location of recitation class or updated office hours, will also be posted in Canvas.
- Feishu: communications

Course Information

- Instructor: Xiaodong Wei
- xiaodong.wei@sjtu.edu.cn, Room 425
- Lab sessions (Starting from Sep 30, PRESENCE REQUIRED): you will be asked to pick one of the two sessions; see Canvas for time and location.
  - Monday (from Sep 30), 12:30-15:40, DZY3-406
  - Monday (from Sep 30), 18:20-21:30, DZY2-303
- Instructor's office hours (starting from Oct 9): Wednesday, 12:30-13:30

Course Information – Two Make-up Classes

- Sep 29 (Sun): online, 16:00-17:40
  - Feishu meeting link: https://vc.feishu.cn/j/633317954
- Oct 12 (Sat): on-site, 16:00-17:40, DXY305

TA Information

- Teaching assistants:
  - Jinbin Zhang, moranxuege@sjtu.edu.cn
  - Miao Qi, qm1713@sjtu.edu.cn
- Recitation classes (mandatory): please see Canvas (starting from week 4)
- TA office hours: please see Canvas (starting from week 4)

References – Not Required

- MATLAB help
- The C Programming Language, 2nd Ed. / Kernighan and Ritchie
- C++ Primer, 5th Ed. / Lippman, Lajoie, and Moo
- You are encouraged to search the Internet to learn, as what I usually do.

What is the Course About?

- The course will focus on:
  - algorithm and problem solving
  - beauty of computer programming
- We will also talk a little about object oriented feature of the programming (more advanced topic can be found in VE280)

What You Will Learn?

- Design algorithms for clearly specified tasks.
- Successfully implement algorithms in MATLAB.
- Successfully implement algorithms in C and C++.
- Be prepared to engage in programming for specific engineering tasks.

Course Details

- This course will be divided into three parts: MATLAB (1/3), C (1/3+), C++ (1/3-)
- Work loads:
  - 3 hrs lecture, 3 hrs lab, and 2 hrs recitation class per week.
  - About 8-10 hrs/week on your lab.
  - 7 labs
  - 1 group project (~5 people/team, on C)
  - 2 midterm exams (1st computer-based, 2nd written)
  - Final exam (open everything)

AI Assistants by SJTU

- Web: https://aiedu.sjtu.edu.cn
- Login with your SJTU account.
- Models: deepseek, Tecent, etc.
- Only for teaching purpose, don’t use it for personal or research purpose.

Course Syllabus

- Check it on Canvas
- Might be adjusted later according to course progress

Honor Code Violation

- Lab and course project:
  - Copy source codes from others.
  - Submit codes that are not your original work.
  - Note: AI assistants are allowed, but you need to know details.
- Exam:
  - Give or receive any unauthorized help to/from others.
- Honor Code is taken very seriously in this class, and confirmed violations will be seriously punished according to the JI Honor Code Policy.

What You Can Do

- You are encouraged to help each other understanding the course contents and lab/project requirements.
- You are encouraged to discuss course related problems with your instructor, TAs and other students.
- You can use the code given in classes or lecture notes.

Grading Policy

- Grading policy:
  - Lab: 20% (presence:5%, “finishing” the lab problems on site:15%).
  - Project: 10%.
  - Two midterm exams: 20% for each midterm.
  - Final exam: 30%.
- Late submission is not allowed.
- Exception: valid reason with supporting documents, such as the doctor’s note or hospitalization records. Exceptions will be handled case by case.

More About Honor Code

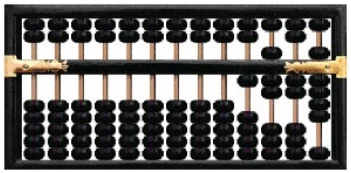
- Look at the grading policy, you can know that, it is not easy to get an A for the class, but it is not easy to fail either.
- So please don’t break the rules, it’s dangerous.
- Honor code violation will be reported, and will be punished (for example, downgrade or fail) if confirmed.

Introduction to Computers and Programming

Video

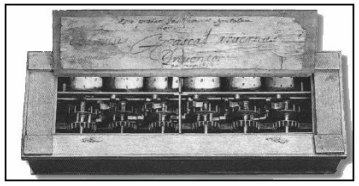
- History Of Programming (Educational Video).mp4
  (https://www.youtube.com/watch?v=tzUbxALPcyw)

A Brief History of Computing

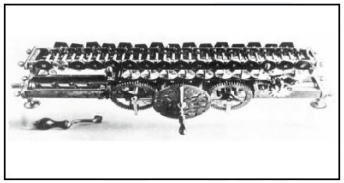


- Although electronic computers are relatively new, mechanical computers are much older. The abacus goes back to almost 4000 years ago in Mesopotamia.
- In the 17th century, several mechanical computing devices were developed in Europe.

Reconstruction of 1623 Wilhelm Schickard machine (Deutsches Museum, Munich)



Blaise Pascal's 1641 "Pascaline" machine (Musée des Arts et Metiers, Paris)



Gottfried Wilhelm von Leibniz's calculating wheel (ca. 1671) (IBM)

- However, the most important conceptual breakthroughs came in the early part of the 19th century . . .

Babbage's Machines

- Charles Babbage is one of the most fascinating figures in the history of computing. Captivated by the idea that he could build a machine to produce mathematical tables, Babbage designed two machines, the Difference Engine and the Analytical Engine, that anticipated many of the features found in modern computers.
- Although Babbage was unable to finish either machine during his lifetime, the Science Museum in London was able to complete a full-scale Difference Engine for the 200th anniversary of his birth.

Charles Babbage (1791-1871)

Ada Byron, The First Programmer

- Augusta Ada Byron, the daughter of the famous English poet, Lord Byron, was encouraged to pursue her interests in science and mathematics at a time when few women were allowed to study those subjects. At the age of 17, Ada met Charles Babbage and became fascinated by his machines. Ada was convinced of the potential of Babbage's Analytical Engine and wrote extensive notes on its design, along with several complex mathematical programs that have led many people to characterize her as the first programmer. In 1980, the U.S. Department of Defense named the programming language Ada in her honor.

Augusta Ada Byron,
Lady Lovelace (1815–1852)

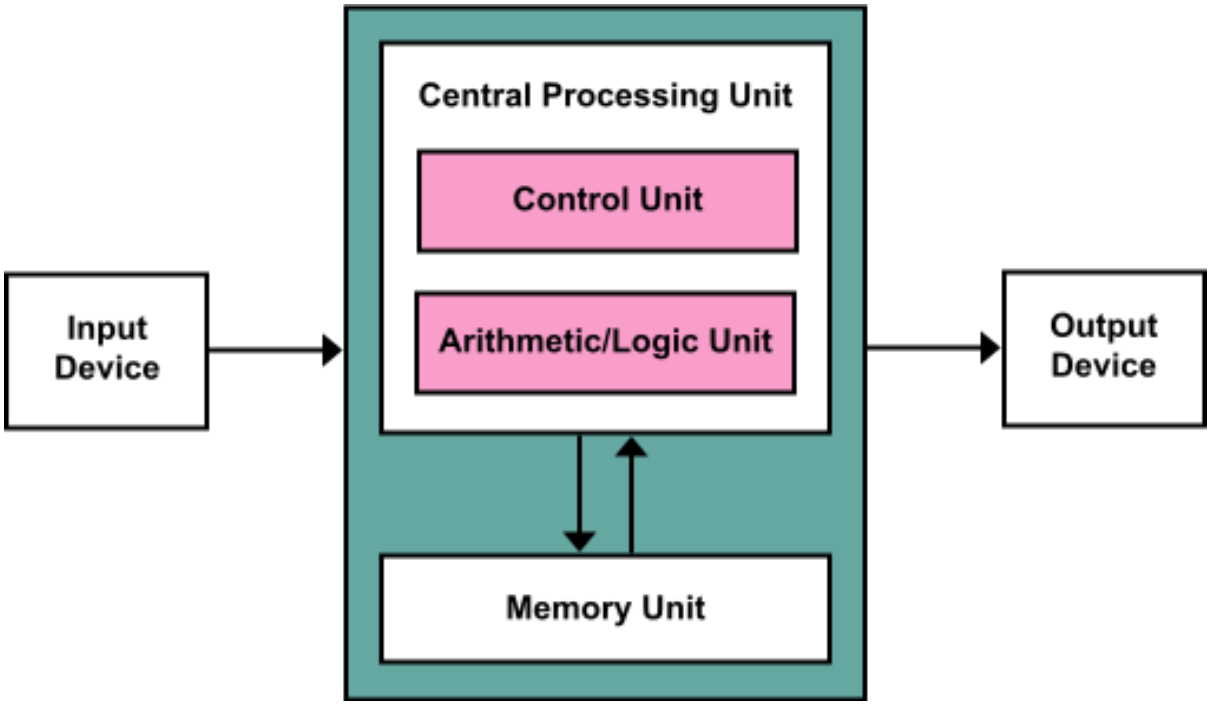
The Birth of Modern Computing

- The question of who invented the modern computers is not an easy one, given the competing claims for that achievement.
- In 1939, John Atanasoff and Clifford Berry built a prototype computer (ABC) at Iowa State and a large machine in 1942.
- The first large-scale computer was the Electronic Numerical Integrator and Computer (ENIAC), completed in 1945 under the direction of J. Presper Eckert and John Mauchly at the Moore School of the University of Pennsylvania.
- Konrad Zuse in Germany (Z1) and the World War II cryptography team in England (Colossus) also built early computers.
- Other important contributions during the early years include stored-programming concept generally attributed to John von Neumann and the use of switching circuits to implement binary arithmetic proposed by Claude Shannon.

What is Computer Science?

- Many people imagine that computer science is the study of computers as artifacts and wonder how that can be a science.
- Computer science has more to do with the study of problem solving in which the solutions happen to use computers.
- Computer science draws on a range of intellectual traditions that includes aspects of mathematics, classical science, and engineering.
- Computer science plays an increasingly important role in other disciplines:
  – Biology. Computers made it possible to map the human genome.
  – Economics. Computers enable the creation of better economic models.
  – Psychology. Artificial intelligence helps us to understand the brain.
  – Environment. Climate models require modern computing technology.
  – Literature. Computerized analysis helps resolve disputed authorship.
  – and most everything else . . .

Computer Architecture



Von Neumann architecture

http://en.wikipedia.org/wiki/Von_Neumann_architecture

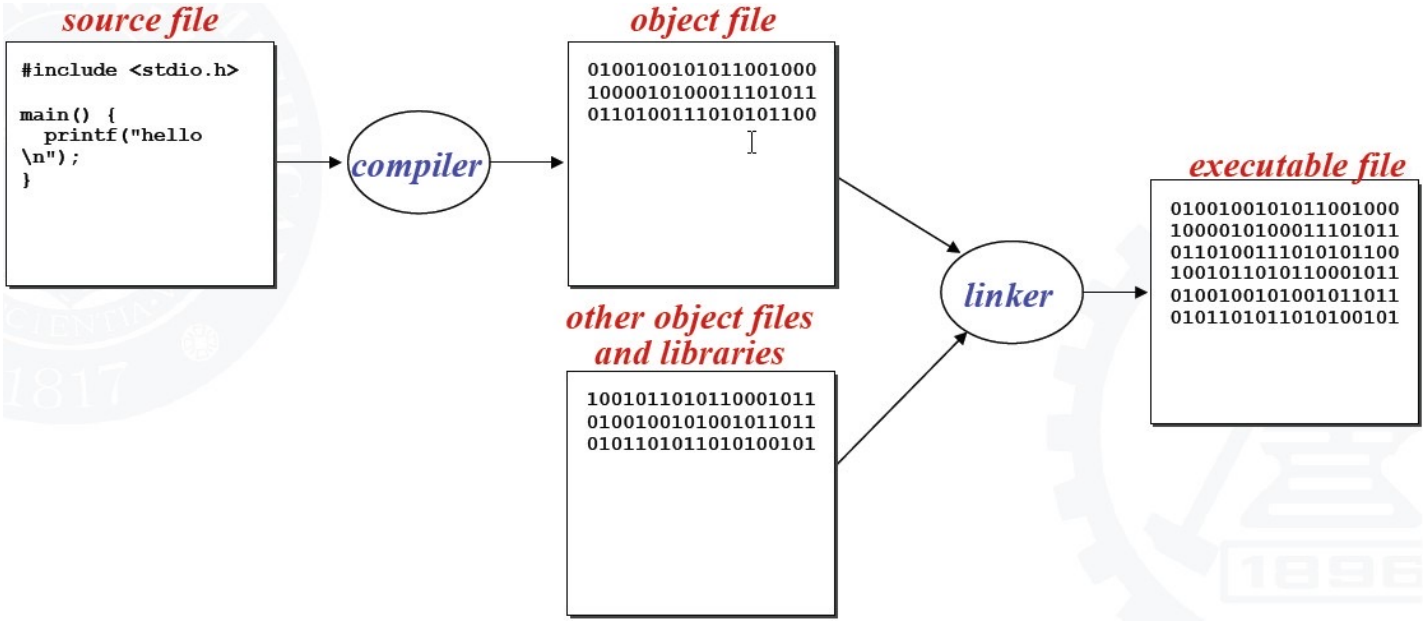
Algorithms

- Much of computer science involves the study of algorithms.
- In an informal sense, you can think of an algorithm as simply a procedure for solving a problem.
- To meet its more formal definition, an algorithm must be:
  - Clearly and unambiguously defined.
  - Effective, in the sense that its steps are executable.
  - Finite, in the sense that it terminates after a bounded number of steps.

Programming Process

- Each computer system understands a low-level language that is specific to that type of hardware, which is called its machine language.
- Programmers typically write their software in a higher level language that is easier for humans to understand.
- To execute a program written in a higher-level language, the computer must adopt one of two strategies:
  - The classical approach is to translate the higher-level language into machine language. This strategy is called compilation.
  - A second approach is to simulate the program operation without actually translating it to machine language. This strategy is called interpretation.

The Compilation Process



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.

Number Systems

- **Binary:**
  0, 1
- **Octal (seldom used now):**
  0, 1, 2, 3, 4, 5, 6, 7
- **Decimal:**
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- **Hexadecimal:**
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Number Systems, Cont.

- Computers use binary numbers internally because storage devices like memory and disk are made to store 0s and 1s. A number or a text inside a computer is stored as a sequence of 0s and 1s. Each 0 and 1 is called a bit, short for **bi**nary digi**t**. The binary number system has two digits, 0 and 1.
- Binary numbers tend to be very long and cumbersome. Hexadecimal numbers are often used to abbreviate binary numbers.

Binary Numbers ➔ Decimals

Given a binary number $b_n b_{n-1} b_{n-2} \ldots b_2 b_1 b_0$
the equivalent decimal value can be computed as

$$b_n \times 2^n + b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \ldots + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$$

10 in binary $1 \times 2^1 + 0$ **= 2 in decimal**

1000 in binary $1 \times 2^3 + 0 \times 2^2 + 0 \times 2 + 0$ **= 8 in decimal**

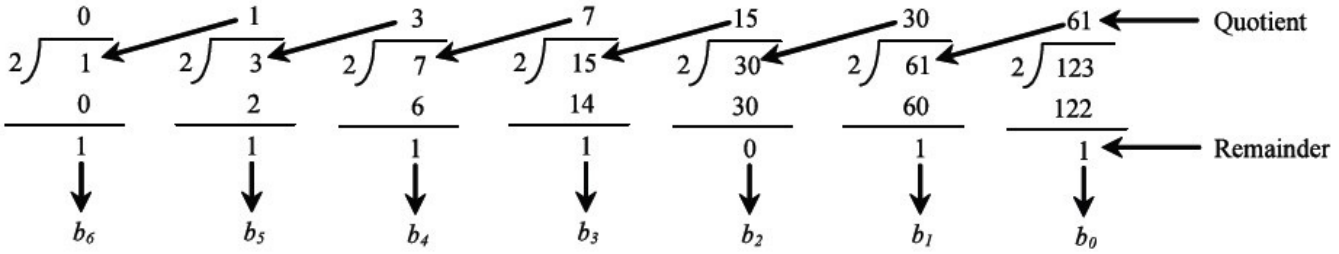
10101011 in binary $1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1$ **= 171 in decimal**

Decimals ➔ Binary

To convert a decimal number d to a binary number is to find the binary digits.. $b_n, b_{n-1}, b_{n-2}, \ldots, b_2, b_1, b_0$ **ch that**

$$d = b_n \times 2^n + b_{n-1} \times 2^{n-1} + b_{n-2} \times 2^{n-2} + \ldots + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$$

These numbers can be found by successively dividing d by 2 until the quotient is 0. The remainders are $b_n, b_{n-1}, b_{n-2}, \ldots, b_2, b_1, b_0$
For example, the decimal number 123 is 1111011 in binary.
The conversion is conducted as follows:



Hexadecimals ➔ Decimals

- **The hexadecimal number system has sixteen digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. The letters A, B, C, D, E, and F correspond to the decimal numbers 10, 11, 12, 13, 14, and 15. Given a hexadecimal number $h_n h_{n-1} h_{n-2} \ldots h_2 h_1 h_0$**

  The equivalent decimal value is

$$h_n \times 16^n + h_{n-1} \times 16^{n-1} + h_{n-2} \times 16^{n-2} + \ldots + h_2 \times 16^2 + h_1 \times 16^1 + h_0 \times 16^0$$

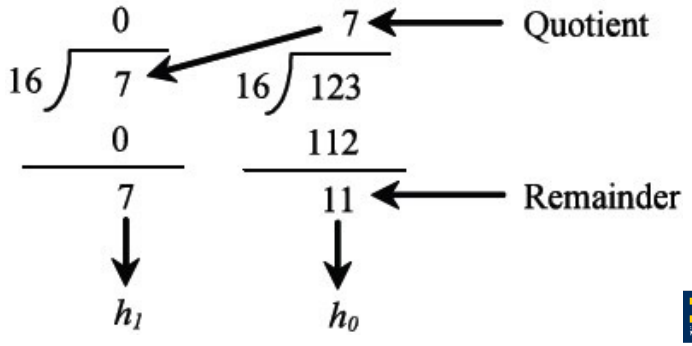
7F in hex $7 \times 16^1 + 15 = 127$ **in decimal**

FFFF in hex $15 \times 16^3 + 15 \times 16^2 + 15 \times 16 + 15 = 65535$ **in decimal**

Decimals ➔ Hexadecimals

- To convert a decimal number d to a hexadecimal number is to find the hexadecimal digits h_n, h_{n-1}, …, h_1, h_0 such that $d = h_n x 16^n + h_{n-1} x 16^{n-1} + \ldots + h_1 x 16^1 + h_0 x 16^0$
- These numbers can be found by successively dividing d by 16 until the quotient is 0. The remainders are h_0, h_1, …, h_{n-1}, h_n

For example, the decimal number 123 is 7B in hexadecimal. The conversion is conducted as follows



Hexadecimal <-> Binary

Binary	Hex	Decimal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	A	10
1011	B	11
1100	C	12
1101	D	13
1110	E	14
1111	F	15

To convert a hexadecimal number to a binary number, simply convert each digit in the hexadecimal number into a four-digit binary number.

To convert a binary number to a hexadecimal, convert every four binary digits from right to left in the binary number into a hexadecimal number. For example,

1 1 1 0 0 0 1 1 0 1

Windows Calculator

- The Windows Calculator is a useful tool for performing number conversions.