CSC1001_T10_T11_T12
Tutorial Notes
September 14, 2023
Kaiyan Zheng (FE)
CUHK(SZ)
kaiyanzheng@link.cuhk.edu.cn

1. **Teaching Fellows: I will introduce the teaching team, including myself, who will be here to support you throughout the course.**
2. **Schedule for Tutorials: We'll discuss the tutorial schedule, so you know when and where to attend these important sessions.**
3. **Installing Python: I'll provide guidance on how to install Python, the primary programming language for this course, on your computer.**
4. **Assignments and Exams: I'll explain the structure of assignments and exams, including deadlines and grading criteria.**
5. **Useful Tips: I'll share some helpful tips and strategies for succeeding in the course and staying on track.**
6. **An Example: Finally, I'll walk you through a simple Python example to give you a taste of what we'll be learning and working on in this course.**

What is a computer programming language? It's the way we communicate with computers.
Your code goes through a series of steps:
Code → ... → Machine instructions → Execution.

In programming, we have low-level languages like assembly, which are challenging to use,
and high-level languages such as C, C++, Python, Java, and C#, which are more user-friendly.
While low-level languages offer execution efficiency,
high-level languages prioritize coding efficiency.
Learning Python this term opens the door to the programming world, so seize this opportunity!

knowledge points of Python

1. Identifier: Identifiers are names used for variables, functions, classes, and other objects in Python.
2. Basic Grammars: Python has a simple and readable syntax with clear indentation rules.
3. Variables' Type: Python has dynamic typing, meaning variable types are determined automatically based on the assigned value.

4. Operator: Python supports a wide range of operators for mathematical and logical operations.
5. Conditional Statement, Loop Statement: You can use if, elif, else for conditions and for/while loops for repetitive tasks.
6. String: Python's string handling is versatile, allowing you to manipulate and format text effectively.
7. List, Tuple, Dictionary: Lists are ordered collections, tuples are immutable, and dictionaries use key-value pairs for data storage.
8. Function: Functions are reusable blocks of code that you can define to perform specific tasks, promoting code modularity and reusability.

9. **Module: Python modules are files containing Python code that can be reused in other programs, making code organization and reusability more manageable.**
10. **File, I/O: Python allows you to read from and write to files, making it useful for data processing and storage.Here, I/O stands for "Input/Output".**
11. **OOP: Object-oriented Programming: Python supports object-oriented programming, where you can create classes and objects to model and structure your code in a more organized and modular way.**

1. Data Representation using Positional Notation:
- Understand how different positional notation systems like binary, octal, and hexadecimal represent data.
- Learn how to convert numbers from binary, octal, and hexadecimal systems to decimal.

2. Conversion from Decimal System (10) to Other Systems:
- Learn to convert decimal numbers to hexadecimal, octal, and binary.
- Cover both the integer and fractional parts of decimal numbers.

3. One-to-One Relationships:
- Explore the one-to-one relationships between octal and binary systems, as well as hexadecimal and binary systems.
- Understand how these different systems relate to each other and how they are used in computing.

These topics will provide students with a solid foundation in data representation and conversions in different positional notation systems,
which is essential in computer science and digital systems.

Data Decomposition

Imagine that numbers are like puzzles, and each digit in a number is a piece of that puzzle. These puzzle pieces have different values depending on where

they're placed. The first piece, all the way to the left, is like piece number '0' in our puzzle. As we move to the right, we increase the piece number by one each time.
Now, we also have a set of special tools called 'weights,' and each piece has its own weight. These weights determine the importance of each piece in the puzzle. The first piece has a weight of '1,' the second piece has a weight of 'base,' the third has a weight of 'base squared,' and so on, where 'base' is the base of our number system.
So, when we want to represent a number, we can think of it as putting together these puzzle pieces.
Each piece is like 'a_i' (where 'i' is the piece number), and we multiply it by the corresponding weight 'b^i.' When we add up all these 'a_i * b^i' for every piece 'i,' we get our complete number. Here n is the number of bases in the system. Like decimal system, n = 10.

This concept applies to all number systems, whether it's binary, decimal, hexadecimal, or any other. It's a bit like building a number from its basic parts, with each part having its own unique value and significance in the final result."

From Others to Decimal
Practice 1
To convert the binary number 100.1 to its decimal equivalent, follow these steps:

1. **Separate the Integer and Fractional Parts:** Divide the binary number into two parts: the integer part (left of the decimal point) and the fractional part (right of the decimal point). In this case, it's 100 (integer) and 1 (fractional).
2. **Convert the Integer Part:**
    - Start from the rightmost digit (least significant digit).
    - Multiply each digit by 2 raised to the power of its position.
    - Sum up the results.

For 100:

- 0 * 2^0 = 0
- 0 * 2^1 = 0
- 1 * 2^2 = 4

Add these values together: 0 + 0 + 4 = 4
So, the integer part in decimal is 4.

3. **Convert the Fractional Part:**
    - Start from the left of the binary point (decimal point).
    - Multiply each digit by 2 raised to the power of its position (negative exponent).
    - Sum up the results.

For 1 (in binary):

- 1 * 2^(-1) = 0.5

So, the fractional part in decimal is 0.5.

4. **Combine the Integer and Fractional Parts:**
   - Add the integer and fractional parts together to get the final decimal value.

4 (integer part) + 0.5 (fractional part) = 4.5

So, the binary number 100.1 is equal to the decimal number **4.5.**

$$(\mathbf{100.1})_{\mathbf{8}} = \mathbf{1} \times \mathbf{8^2} + \mathbf{0} \times \mathbf{8^1} + \mathbf{0} \times \mathbf{8^0} + \mathbf{1} \times \mathbf{8^{-1}} = (\mathbf{64.125})_{\mathbf{10}}$$

To convert the hexadecimal number 100.1 to its decimal equivalent, follow these steps:

1. **Separate the Integer and Fractional Parts:** Divide the hexadecimal number into two parts: the integer part (left of the decimal point) and the fractional part (right of the decimal point). In this case, it's 100 (integer) and 1 (fractional).
2. **Convert the Integer Part:**
   - Start from the rightmost digit (least significant digit).
   - Multiply each digit by 16 raised to the power of its position.
   - Sum up the results.

For 100:

- 0 * 16^0 = 0
- 0 * 16^1 = 0
- 1 * 16^2 = 256

Add these values together: 0 + 0 + 256 = 256

So, the integer part in decimal is 256.

3. **Convert the Fractional Part:**
   - Start from the left of the hexadecimal point (decimal point).
   - Multiply each digit by 16 raised to the power of its position (negative exponent).
   - Sum up the results.

For 1 (in hexadecimal):

- 1 * 16^(-1) = 0.0625

So, the fractional part in decimal is 0.0625.

4. **Combine the Integer and Fractional Parts:**
   - Add the integer and fractional parts together to get the final decimal value.

256 (integer part) + 0.0625 (fractional part) = 256.0625

So, the hexadecimal number 100.1 is equal to the decimal number **256.0625.**

Practice 2

1. **Separate the Integer and Fractional Parts:** Divide the binary number into two parts: the integer part (left of the binary point) and the fractional part (right of the binary point). In this case, it's 10101010 (integer) and 0101 (fractional).
2. **Convert the Integer Part:**
    - Start from the rightmost digit (least significant digit).
    - Multiply each digit by 2 raised to the power of its position.
    - Sum up the results.

For 10101010:

- 0 * 2^0 = 0
- 1 * 2^1 = 2
- 0 * 2^2 = 0
- 1 * 2^3 = 8
- 0 * 2^4 = 0
- 1 * 2^5 = 32
- 0 * 2^6 = 0
- 1 * 2^7 = 128

Add these values together: 0 + 2 + 0 + 8 + 0 + 32 + 0 + 128 = 170
So, the integer part in decimal is **170.**

3. **Convert the Fractional Part:**
    - Start from the left of the binary point.
    - Multiply each digit by 2 raised to the power of its position (negative exponent).
    - Sum up the results.

For 0101 (in binary):

- 0 * 2^(-1) = 0
- 1 * 2^(-2) = 0.25
- 0 * 2^(-3) = 0
- 1 * 2^(-4) = 0.0625

Add these values together: 0 + 0.25 + 0 + 0.0625 = 0.3125
So, the fractional part in decimal is 0.3125.

4. **Combine the Integer and Fractional Parts:**
    - Add the integer and fractional parts together to get the final decimal value.

170 (integer part) + 0.3125 (fractional part) = 170.3125
So, the binary number 10101010.0101 is equal to the decimal number **170.3125.**

1. Octal Number 6.6:
Integer Part: 6 (in octal) = 6 (in decimal)
Fractional Part: 0.6 (in octal) = 6/8 (in decimal) = 0.75
So, the decimal equivalent is **6.75.**

2. Octal Number 66.66:
Integer Part: 66 (in octal) = (6 * 8^1) + (6 * 8^0) = 48 + 6 = 54 (in decimal)
Fractional Part: 0.66 (in octal) = (6/8 * 8^(-1)) + (6/8 * 8^(-2)) = 0.75 + 0.09375 = 0.84375
So, the decimal equivalent is **54.84375.**
3. Octal Number 666.666:
Integer Part: 666 (in octal) = (6 * 8^2) + (6 * 8^1) + (6 * 8^0) = 384 + 48 + 6 = 438 (in decimal)
Fractional Part: 0.666 (in octal) = (6/8 * 8^(-1)) + (6/8 * 8^(-2)) + (6/8 * 8^(-3)) = 0.75 + 0.09375 + 0.01171875 = 0.85546875
So, the decimal equivalent is 438.85546875.

1. Convert the Integer Part:

- 3D5 (in hexadecimal) is equivalent to:
  - (5 * 16^0) + (D * 16^1) + (3 * 16^2)
  - (5 * 1) + (13 * 16) + (3 * 256)
  - 5 + 208 + 768
  - 981 (in decimal)

2. Convert the Fractional Part: （omited）

$$(3D5.F9)_{16} = 3 \times 16^2 + 13 \times 16^1 + 5 \times 16^0 + 15 \times 16^{-1} + 9 \times 16^{-2}$$
$$= (981.97265625)_{10}$$

To represent the year 2019 in different bases:

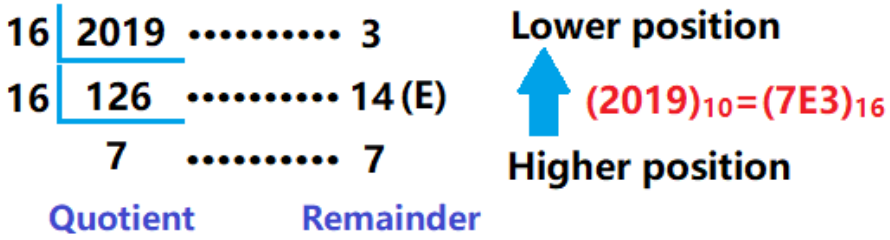
- In hexadecimal (base 16): 2019 can be represented as $\boldsymbol{Year}$ (7E3)_16, where 7E3 is the hexadecimal equivalent of 2019.
- In octal (base 8): 2019 can be represented as $\boldsymbol{Year}$ (3737)_8, where 3737 is the octal equivalent of 2019.
- In binary (base 2): 2019 can be represented as $\boldsymbol{Year}$ (11111100011)_2, where 11111100011 is the binary equivalent of 2019.

So, you have three different representations of the year 2019 in hexadecimal, octal, and binary bases.

Let's break down the steps to represent the year 2019 in different bases:
1. Hexadecimal (Base 16):

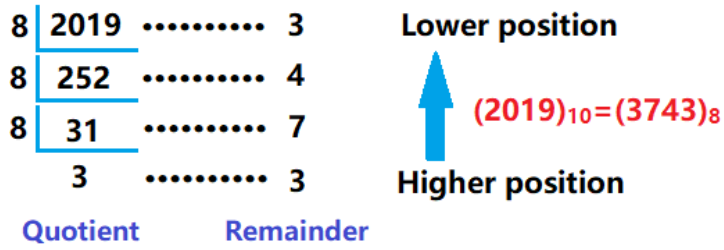
- Start with the decimal number 2019.
- Divide 2019 by 16 (the base for hexadecimal):
  - 2019 ÷ 16 = 126 with a remainder of 3. This remainder is the least significant digit in hexadecimal.

- Now, divide the quotient (126) by 16 again:
  - 126 ÷ 16 = 7 with a remainder of 14. In hexadecimal, 14 is represented as 'E'.
- Finally, divide the new quotient (7) by 16:
  - 7 ÷ 16 = 0 with a remainder of 7. In hexadecimal, 7 remains as '7'.
- Read the remainders in reverse order to get the hexadecimal representation:
  - 2019 in hexadecimal is 7E3. So, ***Year*** (7E3)_16 represents the year 2019.



2. Octal (Base 8):

- Start with the decimal number 2019.
- Divide 2019 by 8 (the base for octal):
  - 2019 ÷ 8 = 252 with a remainder of 3. This remainder is the least significant digit in octal.
- Now, divide the quotient (252) by 8 again:
  - 252 ÷ 8 = 31 with a remainder of 4. In octal, 4 remains as '4'.
- Finally, divide the new quotient (31) by 8:
  - 31 ÷ 8 = 3 with a remainder of 7. In octal, 7 remains as '7'.
- Read the remainders in reverse order to get the octal representation:
  - 2019 in octal is 3743. So, ***Year*** (3743)_8 represents the year 2019.



3. Binary (Base 2):

- Start with the decimal number 2019.
- Divide 2019 by 2 (the base for binary):
  - 2019 ÷ 2 = 1009 with a remainder of 1. This remainder is the least significant digit in binary.
- Now, divide the quotient (1009) by 2 again:
  - 1009 ÷ 2 = 504 with a remainder of 1. In binary, 1 remains as '1'.
- Continue dividing by 2 until the quotient becomes 0:
  - 504 ÷ 2 = 252 with a remainder of 0.
  - 252 ÷ 2 = 126 with a remainder of 0.

    - 126 ÷ 2 = 63 with a remainder of 0.
    - 63 ÷ 2 = 31 with a remainder of 1.
    - 31 ÷ 2 = 15 with a remainder of 1.
    - 15 ÷ 2 = 7 with a remainder of 1.
    - 7 ÷ 2 = 3 with a remainder of 1.
    - 3 ÷ 2 = 1 with a remainder of 1.
    - 1 ÷ 2 = 0 with a remainder of 1.
- Read the remainders in reverse order to get the binary representation:
    - 2019 in binary is 11111100011. So, $Year$ (11111100011)_2 represents the year 2019.

Practice 4

Convert decimal number 100.96875 into binary, octal and hexadecimal number.

1. Binary (Base 2):

Integer Part (Before Decimal Point):

- Convert the integer part of 100 to binary: 100 (decimal) = 1100100 (binary).

Fractional Part (After Decimal Point):

- Convert the fractional part of 0.96875 to binary:
    - Multiply the fractional part by 2.
    - The whole number part of the result is the next binary digit.
    - Continue this process until you get the desired precision or until the fractional part becomes 0.

0.96875 * 2 = 1.9375 (binary: 1) 0.9375 * 2 = 1.8750 (binary: 1) 0.8750 * 2 = 1.7500 (binary: 1) 0.7500 * 2 = 1.5000 (binary: 1) 0.5000 * 2 = 1.0000 (binary: 1)

- Combine the binary representations of the integer and fractional parts:
    - 100.96875 in binary is 1100100.1111.

2. Octal (Base 8):

- Convert the integer part of 100 to octal: 100 (decimal) = 144 (octal).
- Convert the fractional part of 0.96875 to octal:
    - Similar to the binary conversion, multiply the fractional part by 8 and record the whole number part.

0.96875 * 8 = 7.75 (octal: 7) 0.75 * 8 = 6.00 (octal: 6)

- Combine the octal representations of the integer and fractional parts:
    - 100.96875 in octal is 144.76.

3. Hexadecimal (Base 16):

- Convert the integer part of 100 to hexadecimal: 100 (decimal) = 64 (hexadecimal).
- Convert the fractional part of 0.96875 to hexadecimal:
    - Similar to the binary and octal conversions, multiply the fractional part by 16 and record the whole number part.

0.96875 * 16 = 15.5 (hexadecimal: F) 0.5 * 16 = 8.0 (hexadecimal: 8)

- Combine the hexadecimal representations of the integer and fractional

parts:
    - 100.96875 in hexadecimal is 64.F8.

So, the decimal number 100.96875 can be represented as:

- In binary: 1100100.1111
- In octal: 144.76
- In hexadecimal: 64.F8

For exam,
Put the table like this in your cheating paper!

$$(0)_8 = (000)_2$$
$$(1)_8 = (001)_2$$
$$(2)_8 = (010)_2$$
$$(3)_8 = (011)_2$$
$$(4)_8 = (100)_2$$
$$(5)_8 = (101)_2$$
$$(6)_8 = (110)_2$$
$$(7)_8 = (111)_2$$

1. Binary (Base 2) to Octal (Base 8) and Hexadecimal (Base 16):

- Binary 10100101.01011:
    - Octal: 245.26
    - Hexadecimal: A5.58

2. Octal (Base 8) to Binary (Base 2) and Hexadecimal (Base 16):

- Octal 7654.123:
    - Binary: 111110101100.001010011
    - Hexadecimal: FAC.298

➢1.Which of the following inequalities is correct?

A. $(0.111)_2 < (0.777)_{10}$ B.$(0.78)_{10} > (0.C)_{16}$

C.$(0.6)_{10} > (0.AB)_{16}$ D.$(0.101)_2 < (0.A)_{16}$

➢2.Which of the following equalities is incorrect?

A.$(0.875)_{10} = (0.E)_{16}$ B. $(11.4)_8 = (9.8)_{16}$

C.$(10)_{10} = (1010)_2$ D. $(101111)_2 = (AF)_{16}$

0.111 in binary is equal to:
(1 * 2^(-1)) + (1 * 2^(-2)) + (1 * 2^(-3))
= (1/2) + (1/4) + (1/8)
= 4/8 + 2/8 + 1/8
= 7/8 =0.875>0.777

"C" in hexadecimal is equivalent to 12 in decimal.
So, 0.C in decimal is equal to 0 + 12/16 = 12/16 = 3/4 = 0.75 in decimal.

0.AB in decimal is equal to:

(10/16) + (11/16^2)

= 10/16 + 11/256

= 160/256 + 11/256

= (160 + 11)/256

= 171/256
= 0.66796875 >0.6

"A" in hexadecimal is equivalent to 10 in decimal.

So, 0.A in decimal is equal to:

10/16

= 5/8

= 0.625 in decimal.

0.E in decimal is equal to:
14/16
= 7/8
= 0.875 in decimal.