



UNIT 1: INTRODUCTION TO COMPUTATIONAL SYSTEMS

NOTES — SHORT QUESTIONS ONLY

1.1 THEORY OF SYSTEMS

1**Define a system.**

A system is an organized collection of interdependent parts that work together to perform a specific function or achieve a common goal.

★ EXERCISE

Page 2

2**Define an Information System.**

An Information System is a combination of technology and people that collects, processes, stores, and shares data to help an organization make decisions and operate effectively. It is simply an organized set of components that are coordinated to perform a designated function. All the components of the system are in some way related to each other and the functioning of the other components enhances the operation of the system.

◆ MAIN TEXT

Page 2

3**What are the four fundamental concepts that describe any system?**

A system is described by its objectives, components, communication among components and environment in which it works. The components of a system communicate with each other to achieve the system's objective in an environment.

◆ MAIN TEXT

Page 3

4**What is meant by the 'Objective' of a system? Give two examples.**

Every system has a purpose or goal that it wishes to fulfil. Analyzing a system's operation requires understanding its aim. This insight improves the efficiency and efficacy of the present system.

Examples (from textbook):

- A transport system aims to transfer people and products securely and effectively between locations.
- A computer system's principal goal is to process data and provide useful information to users.

◆ MAIN TEXT

Page 3

5**Define 'Components' in a system and explain their significance.**

Components are the building blocks of any system. Each component plays a specific role and contributes to the overall functionality of the system. Understanding the role of each component of the system is essential to understand how the entire system works. This helps in identifying problems, improving performance, and refining system design. Smooth and proper working of these components together ensures the system meets its objectives.

◆ MAIN TEXT

Page 3

6**Define the 'Environment' of a system and how it affects system operation.**

The environment of a system includes everything external to the system that interacts with it. It consists of all external factors that affect the system's operation. Understanding the environment of a system is important as it influences the system's performance and behavior by providing inputs and receiving outputs. Intelligent systems adjust to changes in their environment to continue their functionality.

◆ MAIN TEXT

Page 3

7**Why is communication and interaction among system components necessary?**

Communication and interaction among system components is key to the functioning of a system. It ensures that components work together in an organized and smooth manner to achieve the system's objectives. For example, in a computing system the CPU communicates with memory to fetch and store data, and in a biological system brain sends signals to muscles to initiate movement.

◆ MAIN TEXT

Page 3

8**Differentiate between a simple system and a complex system with examples.**

Systems can be simple, like a thermostat, or complex, like the human body or a computer network.

- Simple System: Has straightforward components and predictable interactions (e.g., a thermostat regulating temperature).
- Complex System: Consists of many interacting subsystems with intricate relationships (e.g., the human body, a computer network).

◆ MAIN TEXT

Page 3

9 Describe a computer as a system (depicted in Figure 1.1).

◆ MAIN TEXT

Every computer, machine, or device has components that work together to achieve an outcome. As depicted in Figure 1.1, a Computer System consists of:

- System Unit
- Input Device (Keyboard, Mouse)
- Output Devices (Monitor, Printer)
- Mass Storage Device.

Page 2

1.2 SOFTWARE

10 Define software and state what happens without it.

◆ MAIN TEXT

Software is a collection of programs and instructions that tell a computer what to do and how to do. Without software, computers would be useless machines.

Page 4

11 What was the 'Creeper' virus, when was it created, and what message did it display?

● DID YOU KNOW?

The first computer virus, called "Creeper," was created in 1971 as an experimental self-replicating program. It simply displayed the message:
"I'm the creeper, catch me if you can!"

Page 4

12 Define system software and provide two examples.

★ EXERCISE

System software is designed to manage the system resources and provide a platform for application software to run. It acts as a bridge between the hardware and the user applications.

Examples include:

1. Operating Systems: Microsoft Windows, macOS, Linux.
2. Device Drivers: Printer drivers, graphics card drivers, sound card drivers.

Page 4

13 Name three major types of system software with examples.

◆ MAIN TEXT

- Operating Systems: Examples include Microsoft Windows, macOS, and Linux.
- Device Drivers: These include printer drivers, graphics card drivers, and sound card drivers.
- Utility Programs: Examples are antivirus software, disk cleanup tools, and backup software.

Page 4

14 Define device drivers and give their importance.

◆ MAIN TEXT

Device drivers are system software that include printer drivers, graphics card drivers, and sound card drivers. They allow the operating system to communicate with and control specific hardware devices.

Page 4

15 Define utility programs and give two examples.

◆ MAIN TEXT

Utility programs are system software used to enhance system performance and security. Examples are antivirus software, disk cleanup tools, and backup software.

Page 4

16 Define application software and give two examples.

◆ MAIN TEXT

Application software is designed to help users perform specific tasks. These programs are built to fulfill user needs and are typically more varied than system software.

Examples: Microsoft Word, Google Chrome, VLC Media Player.

Page 4

17 Name four common categories of application software with examples.

◆ MAIN TEXT

- Word Processors: Such as Microsoft Word and Google Docs.
- Web Browsers: Such as Google Chrome, Mozilla Firefox, and Safari.
- Games: Such as Minecraft, Fortnite, and Among Us.
- Media Players: Such as VLC Media Player and Windows Media Player.

Page 5

18 Differentiate between system software and application software.

★ EXERCISE

- Purpose: System software manages and operates computer hardware, making it possible for application software to run. Application software helps the user to perform specific tasks.
- Examples: System software includes operating systems and device drivers. Application software includes word processors, web browsers, and games.
- Installation: System software is usually pre-installed on a computer, while application software can be installed by the user as needed.

Page 5

19 Why should system software be kept updated?

💡 TIDBITS

Always keep your system software updated to ensure your computer runs smoothly and is protected from security threats.

Page 5

20 What is the Von Neumann architecture?

◆ MAIN TEXT

Page 5

The Von Neumann architecture is a computer paradigm that describe a system in which the hardware of the computer has four primary components: the memory, the Central Processing Unit (CPU), input mechanisms, and output mechanisms. This model is called the John von Neumann model, the Neumann model named in honor of the mathematician and physicist who contributed to its development during the 1940s.

21 Enlist the main components of the Von Neumann architecture.

★ EXERCISE

Page 5

The four primary components that constitute the architecture of the von Neumann computer are:

1. Memory
2. Central Processing Unit (CPU)
3. Input Devices / Mechanisms
4. Output Devices / Mechanisms.

22 Describe the role of Memory (RAM) in the Von Neumann architecture.

◆ MAIN TEXT

Page 6

Memory: Contains both input data and the instructions (program) required for CPU processing. For instance, consider the RAM of your computer: when a program starts it is loaded into RAM to enable faster execution compared to when it runs from the hard disk.

23 Define the Central Processing Unit (CPU) and its two main components.

◆ MAIN TEXT

Page 6

Central Processing Unit (CPU): Performs addition and subtraction, and executes commands provided by the memory. The system has two main components: the Arithmetic Logic Unit (ALU) and the Control Unit (CU).

24 Define Arithmetic Logic Unit (ALU) and state its function.

◆ MAIN TEXT

Page 6

The Arithmetic Logic Unit (ALU) performs mathematical computations and logical operations.

25 Define Control Unit (CU) and state its function.

◆ MAIN TEXT

Page 6

A Control Unit (CU) is a peripheral that governs the activities of the CPU by instructing the ALU and memory to execute tasks according to the program instructions. It ensures the proper and timely execution of duties by all the other components.

26 Illustrate the roles of ALU and CU when calculating $2 + 2$.

◆ MAIN TEXT

Page 6

When doing the calculation $2 + 2$ on a calculator application, the Arithmetic Logic Unit (ALU) handles the numerical values while the control Unit (CU) supervises the whole procedure.

27 What are input devices and output devices? Give examples.

◆ MAIN TEXT

Page 6

- Input Devices: Enable users to input data and instructions into the computer system. Illustrative examples include keyboard, mouse, and microphone. Entering text on the keyboard transmits data to the CPU for subsequent processing.
- Output Devices: Present or communicate the outcomes of the tasks executed by the computer. Consider, for instance, a monitor and printer. Upon completion of data processing, the CPU transmits the outcome to the monitor for visual display.

28 Define System Bus and explain Data Bus, Address Bus, and Control Bus.

◆ MAIN TEXT

Page 6

A System Bus is a communication mechanism that facilitates the movement of data between components inside a computational system. It comprises:

- Data Bus: Transports data.
- Address Bus: Maintains data destination information.
- Control Bus: Transports control electrical signals.

29 What is a key advantage of the Von Neumann architecture?

★ EXERCISE

Page 6

A key advantage of the Von Neumann architecture is its single memory store for both program instructions and data, which simplifies computer design, enables faster execution (as instructions load into RAM), and provides flexible memory utilization.

30 Enlist the four main steps in the Von Neumann architecture's instruction cycle.

★ EXERCISE

Page 7

The Von Neumann architecture encompasses four essential stages for a CPU to carry out instructions, namely:

1. Fetching
2. Decoding
3. Execution
4. Storing.

31	Describe the 'Fetching' stage and the roles of PC and IR. Fetching: The central processing unit retrieves an instruction from the computer's memory. This instruction specifies the operation to be executed by the CPU. • Comprising Components: Memory, CPU, Program Counter (PC), Instruction Register (IR). • Description: The Program Counter (PC) stores the memory address of the subsequent instruction. Once the address is stored in memory, the instruction located at that location is retrieved and placed into the Instruction Register (IR).	◆ MAIN TEXT Page 7
32	Describe the 'Decoding' stage and define Opcode. Decoding: In order to determine the necessary action, the Control Unit (CU) decodes the instruction. • Comprising Components: Control Unit (CU). • Description: The control unit (CU) decodes the opcode (operation code) of the instruction and determines the required procedures and data.	◆ MAIN TEXT Page 7
33	Describe the 'Execution' and 'Storing' stages. • Execution: The CPU processes the instruction. When the instruction involves a computation, it is executed by the Arithmetic Logic Unit (ALU). Any task that requires transferring data between several locations is managed by the CU. Comprising Components: ALU, CU. Description: The Arithmetic and Logic Unit (ALU) carries out mathematical and logical calculations, while the Control Unit (CU) handles data transmission activities. • Storing: The outcome of the computation is either returned to memory or sent to an output device. Comprising Components: Memory and Output Device. Description: The outcome is either stored in a designated memory location or sent to an output device, such as a display.	◆ MAIN TEXT Page 7

1.4 NUMBER SYSTEMS

34	Define a Number System and state why number systems are essential in computing. Number systems are essential in computing because they form the basis for representing, storing, and processing data. Different number systems help computers perform tasks like calculations, data storage, and data transfer. These systems allow computers to represent various kinds of information, such as text, colors, and memory locations.	◆ MAIN TEXT Page 8
35	Define the Decimal Number System with an example. The decimal number system is a base-10 number system that consists of digit from 0 to 9 and we use it in everyday life. That's why each digit of the number represents a power of 10. In the decimal system the place values starting from the rightmost digits are 10^0 , 10^1 , 10^2 , and so on. For example, the decimal number 523 means: $5 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 = 500 + 20 + 3 = 523$.	◆ MAIN TEXT Page 8
36	Define the Binary Number System and state why computers work in binary. In binary, the place values are arranged from the right to left, starting with 2^0 , and ending at 2^n , where each position represents a power of 2. Why computers work in binary: Computers work in binary system especially because this method fits well with electronics. Digital circuits have two states: They can be either on or off. These states are easily represented by the binary digits: 1 represent ON, and 0 represents OFF. When typing on the keyboard, the computer translates every letter to a binary. Similarly, number, text, images, and sound are all, at their lowest level, reduced to binary.	◆ MAIN TEXT Page 8
37	Convert binary number 1011_2 to decimal. $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 0 + 2 + 1 = 11_{10}$.	◆ MAIN TEXT Page 8
38	State the algorithm to convert a decimal number to binary. The following algorithm converts a decimal number to binary: 1. To convert decimal number to binary form, divide the decimal number by 2. 2. Record the remainder. 3. Divide the number by 2 until the quotient which is left after division is 0. 4. Meaning it is represented by the remainders and it's read from the bottom to the top for the binary number.	◆ MAIN TEXT Page 8

39 Convert decimal number 83 to binary (Figure 1.3).

$83 / 2 = 41$ remainder 1
 $41 / 2 = 20$ remainder 1
 $20 / 2 = 10$ remainder 0
 $10 / 2 = 5$ remainder 0
 $5 / 2 = 2$ remainder 1
 $2 / 2 = 1$ remainder 0
 $1 / 2 = 0$ remainder 1

If the remainders are read from bottom to top then it gives the required result in binary, which is 1010011_2 .

40 Define the Octal Number System with an example.

Octal is a positional numeral system with base eight, which implies that a digit to be used ranges from 0 to 7. The last digit is a single digit power of 8 while the other digits are the coefficients. In the octal system, the place values starting from the 8^0 , 8^1 , 8^2 and so on.

For example, the octal number 157 means:

$$1 \times 8^2 + 5 \times 8^1 + 7 \times 8^0 = 64 + 40 + 7 = 111_{10}.$$

41 How many binary bits represent one octal digit? Explain why.

Each octal digit represents three binary digits (bits) because the octal system is base-8, and the binary system is base-2. This relationship arises from the fact that 8 is a power of 2 ($8 = 2^3$). So, each octal digit can be precisely represented by three binary digits (bits). This means that any value from 0 to 7 in octal can be converted into a 3-bit binary number.

42 Convert binary number 110101011_2 into octal.

This number can be divided into groups of three bits from right to left:

110 101 011

Each group of three bits corresponds to a single octal digit:

$110 = 6$

$101 = 5$

$011 = 3$

So, the binary number 110101011 is equal to 653 in octal (653_8).

43 State the algorithm to convert decimal to octal and convert 83 to octal.

Algorithm:

1. To convert the decimal number to an equivalent octal number, divide the number by 8.
2. Write down the remainder.
3. After that divide the obtained quotient by 8.
4. Continue the divisions until one of the numbers results in 0.
5. Octal is a base eight number and the octal number is the remainder read from the bottom up to the top.

Example (Convert 83 to octal):

- $83 / 8 = 10$ remainder 3
- $10 / 8 = 1$ remainder 2
- $1 / 8 = 0$ remainder 1

Going up from bottom, the remainder reading will give the desired result, that is 123 in the octal system (123_8).

44 Why was the octal system used in early computing systems like PDP-8?

The octal system was used in early computing systems like PDP-8. It was used because it is easier to convert between octal and binary than between decimal and binary.

45 Convert decimal numbers 45, 128, and 64 to octal (Class Activity).

- $45 / 8 = 5$ remainder 5; $5 / 8 = 0$ remainder 5 -> Reading bottom-to-top: 55_8
- $128 / 8 = 16$ remainder 0; $16 / 8 = 2$ remainder 0; $2 / 8 = 0$ remainder 2 -> 200_8
- $64 / 8 = 8$ remainder 0; $8 / 8 = 1$ remainder 0; $1 / 8 = 0$ remainder 1 -> 100_8 .

46 Convert octal numbers 57, 124, and 301 to decimal (Class Activity).

- $57_8 = (5 \times 8^1) + (7 \times 8^0) = 40 + 7 = 47_{10}$
- $124_8 = (1 \times 8^2) + (2 \times 8^1) + (4 \times 8^0) = 64 + 16 + 4 = 84_{10}$
- $301_8 = (3 \times 8^2) + (0 \times 8^1) + (1 \times 8^0) = 192 + 0 + 1 = 193_{10}$.

47 Define the Hexadecimal Number System and state what digits and alphabets it uses.
The hexadecimal is a base 16 number system with digit number from 0 to 9 and alphabets from A to F; each digit represents the power of 16 to show the position of the digit. The letter A to F stand for the numeric value of 10 to 15. The digits in hexadecimal move from right to left in place value that are 16^0 , 16^1 , 16^2 ... anothers.

48 Convert hexadecimal number $1A3_{16}$ to decimal.
 $1 \times 16^2 + A \times 16^1 + 3 \times 16^0 = 1 \times 256 + 10 \times 16 + 3 \times 1 = 256 + 160 + 3 = 419_{10}$.

49 Why is hexadecimal used if it is not directly used by computers?
The hexadecimal number system is not directly used by computers either. However, it provides an even more compact representation than octal. This makes it easier for us to read and write large binary numbers. This is because the hexadecimal system is base-16 and the binary system is base-2, therefore every single hexadecimal digit equals four binary bits ($16 = 2^4$). Any hexadecimal number between 0 and 15 can be converted into 4-bit binary number.

50 Convert binary number 1101011010110010_2 into hexadecimal.
Each group of four bits corresponds to a single hexadecimal digit (from right to left):
1101 0110 1011 0010
 $1101 = D$
 $0110 = 6$
 $1011 = B$
 $0010 = 2$
Therefore, the binary number 1101011010110010 equals to the hexadecimal number $D6B2$ ($D6B2_{16}$).

51 State the algorithm to convert decimal to hexadecimal and convert 2297 to hexadecimal.
Algorithm:
1. Convert the decimal number to an absolute value by dividing it by 16.
2. Record the quotient and the remainder.
3. Continue dividing the quotient by 16 and write down the remainder until the quotient is zero.
4. The hexadecimal number, as you might have guessed, is the remainder read from bottom to top.

Example (Convert 2297 to hexadecimal):
 $2297 / 16 = 143$ remainder 9
 $143 / 16 = 8$ remainder F
 $8 / 16 = 0$ remainder 8
Reading the remainders from bottom to top gives the required result, i.e., $8F9$ in hexadecimal ($8F9_{16}$).

1.5 DATA REPRESENTATION IN COMPUTING SYSTEMS

52 Define Whole Numbers (W) in computing and give examples.
Whole numbers are a set of non-negative integers. They include zero and all the positive integers. Mathematically, the set of whole numbers is:
 $W = \{0, 1, 2, 3, \dots\}$
In computing, whole numbers are often used to represent quantities that can't be negative. Examples include the number of students in a school, a person's age in years, and grades, provided there are no negative figures such as credit point balances.

53 What is the formula for the maximum value of an n-bit whole number? Calculate for 1, 2, and 4 bytes.
If n is the number of bits, the maximum value that can be represented is $2^n - 1$ for examples:
• 1-Byte whole number (8 bits): Maximum value = $2^8 - 1 = 255$
• 2-Byte whole number (16 bits): Maximum value = $2^{16} - 1 = 65,535$
• 4-Byte whole number (32 bits): Maximum value = $2^{32} - 1 = 4,294,967,295$.

54 Define Integers (Z) / Signed Integers and describe the Sign Bit.
Integers extend the concept of whole numbers to include negative numbers. In computer programming, we call them signed integers. The set of integers is represented as:
 $Z = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$
To store both positive and negative values, one bit is reserved as the sign bit (the most significant bit). If the sign bit is ON(1), the value is negative; otherwise, it is positive. Using this system, the maximum positive value that can be stored in a 1 byte signed integer is $(01111111)_2$, which is 127_{10} .

55 How does the number of bits affect the range of integer values?

As the bits available to store a value is $n - 1$, hence the maximum value will be $2^{n-1} - 1$, and the minimum value is -2^{n-1} , where n is the total number of bits.

Examples from textbook:

- 1-Byte signed integer (8 bits): Range is -128 to +127 (-2^7 to $2^7 - 1$)
- 2-Byte signed integer (16 bits): Minimum value = $-2^{15} = -32,768$; Maximum value = $2^{15} - 1 = 32,767$
- 4-Byte signed integer (32 bits): Minimum value = $-2^{31} = -2,147,483,648$; Maximum value = $2^{31} - 1 = 2,147,483,647$.

◆ MAIN TEXT

56 State how the minimum value of an integer is computed with examples for 2 and 4 bytes.

The minimum value is computed using the formula -2^{n-1} , where n is the total number of bits:

- 1-Byte Integer (8 bits): Minimum value = $-2^7 = -128$
- 2-Byte Integer (16 bits): Minimum value = $-2^{15} = -32,768$
- 4-Byte Integer (32 bits): Minimum value = $-2^{31} = -2,147,483,648$.

◆ MAIN TEXT

57 Define 1's Complement of a binary number.

1's Complement is a binary number representation system used to express positive and negative numbers in digital systems. It's one of several methods for representing signed numbers in binary. 1's Complement represents negative numbers in binary by inverting all the bits (changing 0s to 1s and 1s to 0s).

◆ MAIN TEXT

58 Define 2's Complement and state the steps to find it.

To store negative values, computers use a method called 2's complement. To find the 2's complement of a binary number, follow these steps:

1. Invert all the bits (change 0s to 1s and 1s to 0s). (1's complement)
2. Add 1 to the Least Significant Bit (LSB).

◆ MAIN TEXT

59 Convert decimal number -5 to an 8-bit binary number using 2's complement.

1. Start with the binary representation of 5: 00000101_2
 2. Invert all the bits: 11111010_2
 3. Add 1: $11111010_2 + 1_2 = 11111011_2$
- So, -5 in 8-bit 2's complement is 11111011_2 .

1.6 BINARY ARITHMETIC OPERATIONS

◆ MAIN TEXT

60 State the four basic rules of binary addition.

Binary addition follows these simple rules:

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$ (with a carry of 1 to the next higher bit).

★ EXERCISE

61 Add 1100_2 and 1011_2 (Exercise Short Question 10).

For Exercise Q10 ($1100_2 + 1011_2$):

```
11002
+ 10112
-----
101112
```

In this addition:

- $0 + 1 = 1$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$ (with a carry of 1 to the next higher bit)

Result = 10111_2 (23_{10}).

◆ MAIN TEXT

62 Explain how subtraction is carried out in binary arithmetic.

In binary arithmetic, subtraction can also be carried out by adding the 2's complement of the subtrahend to the minuend.

63

Subtract 6 from 9 in Binary / Subtract 0011_2 from 1010_2 (Exercise Short Question 11).

★ EXERCISE

Page 15

Example from textbook (Subtract 6 from 9 in Binary):

Minuend = $9_{10} = 1001_2$ Subtrahend = $6_{10} = 0110_2$

Step 1: Find the Two's Complement of the Subtrahend

• Invert the bits of 0110_2 : Inversion: 1001_2 • Add 1 to the inverted number: $1001_2 + 1_2 = 1010_2 = -6_{10}$

Step 2: Add the Minuend and the Two's Complement of the Subtrahend

 $1001_2 + 1010_2 = 10011_2$

Step 3: Discard the Carry Bit

 $10011_2 \rightarrow$ Discard carry $\rightarrow 0011_2 = 3_{10}$.So, $9 - 6 = 3$.For Exercise Q11 (Subtract 0011_2 from 1010_2):1. 2's complement of subtrahend 0011_2 : Inversion = 1100_2 ; Add 1 = 1101_2 2. Add minuend and 2's complement: $1010_2 + 1101_2 = 10111_2$ 3. Discard carry bit: $0111_2 = 7_{10}$.

64

State the steps to multiply binary numbers and multiply 101_2 by 11_2 .

★ EXERCISE

Page 15

Steps to Multiply Binary Numbers:

1. Write down the binary numbers, aligning them by the least significant bit (rightmost bit).

2. Multiply each bit of the second number by each bit of the first number, similar to the long multiplication method in decimal.

3. Shift the partial results one place to the left for each new row, starting from the second row.

4. Add all the partial results to get the final product.

Example (from textbook):

101

 $\times 11$

101 (This is 101×1)101x (This is 101×1 , shifted left)

1111

So, $101_2 \times 11_2 = 1111_2$.

65

How many binary multiplications does a computer CPU perform per second?

💡 DID YOU KNOW?

The Central Processing Unit (CPU) of a computer performs millions of binary multiplications every second to execute complex instructions and run programs!

Page 16

66

State the steps of binary division and divide 1100_2 by 10_2 .

★ EXERCISE

Page 16

Binary division is similar to decimal division but only involves two digits: 0 and 1. It follows steps like comparing, subtracting, and shifting, similarly to long division in the decimal system.

Steps of Binary Division:

1. Compare: Compare the divisor with the current portion of the dividend.

2. Subtract: Subtract the divisor from the dividend portion if the divisor is less than or equal to the dividend.

3. Shift: Shift the next binary digit from the dividend down to the remainder.

4. Repeat: Repeat the process until all digits of the dividend have been used.

Example: Divide 1100_2 by 10_2 (Step 1: Compare 10 with first two 11, subtract 10 from 11 $\rightarrow$ rem 1)(Step 2: Bring down the next digit 0 $\rightarrow$ 10)(Step 3: Compare 10 with 10, subtract 10 from 10 $\rightarrow$ rem 0)(Step 4: Bring down the next digit 0, no more digits left $\rightarrow$ quotient digit 0)Result: $1100_2 / 10_2 = 110_2$.

67

Solve the binary division problems from Class Activity (Page 17).

🔥 CLASS ACTIVITY

Solutions for Class Activity:

(a) $10101_2 \div 10_2 =$ Quotient 1010_2 , Remainder 1_2 (b) $11100_2 \div 11_2 =$ Quotient 1001_2 , Remainder 1_2 (c) $100110_2 \div 101_2 =$ Quotient 111_2 , Remainder 11_2 .

Page 17

68 Define ASCII and state what the acronym stands for.

◆ MAIN TEXT

Page 17

ASCII is an acronym that stands for American Standard Code for Information Interchange. It is a character encoding standard adopted for representing in devices such as computers and similar systems that use text. Each alphabet, number or symbol is given a code number between 0 and 127 as shown in Table 1.3.

69 What is the primary purpose of the ASCII encoding scheme?

★ EXERCISE

Page 17

ASCII enables different computers and devices to exchange text information reliably. It is an industry standard used to encode text in computers and other devices.

70 How many bits are used in standard ASCII encoding?

◆ MAIN TEXT

Page 17

Standard ASCII uses 7 bits and can represent only 128 characters, with code numbers between 0 and 127.

71 How is the name of our country 'Pakistan' encoded using ASCII?

◆ MAIN TEXT

Page 17

- The ASCII code for an upper case letter "P" is 80.
- The code for letter 'a' in ASCII is 97.
- The ASCII code for the letter 'k' is 107.
- It is interesting to know that the ASCII code for the letter 'i' is 105.
- In the ASCII code system, the letter 's' has a code of 115.
- The code for 't' is 116 in ASCII.
- The ASCII code for "a" is 97.
- The ASCII code for 'n' is 110.

72 What is Extended ASCII and how many characters does it include?

◆ MAIN TEXT

Page 20

While the standard ASCII Table includes 128 characters, there is an extended version that includes 256 characters. This extended ASCII uses 8 bits and includes additional symbols, accented letters, and other characters. However, the original 128 characters are the most commonly used and serves as the basis for text representation in computers.

73 Define Unicode and explain why it was developed.

★ EXERCISE

Page 20

Unicode is an attempt at mapping all graphic characters used in any of the world's writing system. Unlike ASCII, which is limited to 7bits and can represent only 128 characters, Unicode can represent over a million characters through different forms of encodings such as, UTF-8, UTF-16, and UTF-32. UTF is an acronym that stands for Unicode Transformation Format.

74 Define UTF-8 and explain why it is backward-compatible with ASCII.

◆ MAIN TEXT

Page 20

It is a variable-length encoding scheme, meaning it can use a different numbers of bytes (from 1 to 4) to represent a character. UTF-8 is backward compatible with ASCII. It means it can understand and use the older ASCII encoding scheme without any problems. Therefore, if we have a text file written in ASCII, it will work perfectly fine with UTF-8, allowing it to read both old and new texts.

75 How are the letter 'A' and Urdu letter 'ب' represented in UTF-8 and UTF-16?

◆ MAIN TEXT

Page 20

- The letter 'A' in Unicode, represented as, U+0041, is 01000001 in the binary format and occupies 8 bits or 1 byte.
- The Urdu letter 'ب' is represented in Unicode as U+0628; its binary format is 11011000 10101000, means it takes 2 bytes in UTF-8.
- In UTF-16: The letter A is equal to 00000000 01000001 in binary or 65 in decimal (2 bytes).
- In UTF-16: The right Urdu letter 'ب' is represented as 00000110 00101000 in binary, which occupies 2 bytes of memory.

76 Differentiate between UTF-16 and UTF-32.

◆ MAIN TEXT

Page 20

- UTF-16: UTF-16 is another variable character encoding mechanism, although it uses either 2 bytes or 4 bytes per character at most. Unlike UTF-8, it is not compatible with ASCII, meaning it cannot translate ASCII code.
- UTF-32: UTF-32 is a method of encoding that uses a fixed length, with all characters stored in 4 bytes per character. This makes it very simple but at the same time it may look a little complicated when it comes to space usage. Example: Alphabet letter 'A' in UTF-32 is represented in binary as 00000000 00000000 00000000 01000001 which is 4 bytes.

77 How do computers store files like images, audio, and video?

◆ MAIN TEXT

Page 21

All these files (images, audio, and video) are stored as binary data, which means they are represented by sequences of 0s and 1s.

78**Differentiate between HDD, SSD, and Cloud Storage.**

- Hard Disk Drive (HDD): Uses spinning disks to read/write data. They offer large storage capacities.
- Solid State Drive (SSD): Uses flash memory for faster access times and better performance.
- Cloud Storage: Stores files on remote servers accessible via the internet, providing flexibility and backup options.

◆ MAIN TEXT**Page 21****79****List the units of data measurement in order from 1 Byte to 1 Yottabyte.**

Data size is usually expressed in byte and its multiples:

- 1 Byte (B) = 8 Bits
- 1 Kilobyte (KB) = 1024 Bytes
- 1 Megabyte (MB) = 1024 Kilobytes
- 1 Gigabyte (GB) = 1024 Megabytes
- 1 Terabyte (TB) = 1024 Gigabytes
- 1 Petabyte (PB) = 1024 Terabytes
- 1 Exabyte (EB) = 1024 Petabytes
- 1 Zettabyte (ZB) = 1024 Exabytes
- 1 Yottabyte (YB) = 1024 Zettabytes.

● DID YOU KNOW?**Page 21**