

Chapter 01 – Computer Science (Class 10)

Created by NBF StudyHub (YouTube)

1. Explain decimal, binary, octal, and hexadecimal number systems with examples.

Ans:

These are different number systems used in computing:

◆ Decimal Number System

- Base: 10
- Digits: 0–9
- Used by humans in daily life
- Example: 254 (means $2 \times 100 + 5 \times 10 + 4 \times 1$)

◆ Binary Number System

- Base: 2
- Digits: 0 and 1
- Used by computers
- Example: 1010 (means $1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 = 10$ in decimal)

◆ Octal Number System

- Base: 8
- Digits: 0–7
- Used as a shorthand for binary
- Example: $17_8 = 1 \times 8 + 7 \times 1 = 15$ in decimal

◆ Hexadecimal Number System

- Base: 16
- Digits: 0–9 and A–F (A = 10, F = 15)
- Used in memory addresses and color codes
- Example: $1F_{16} = 1 \times 16 + 15 = 31$ in decimal

□ Summary Table:

System	Base	Digits Used	Example
Decimal	10	0–9	254
Binary	2	0, 1	1010
Octal	8	0–7	$17_8 = 15_{10}$
Hexadecimal	16	0–9, A–F	$1F_{16} = 31_{10}$

2. What is 2's complement, and why is it used? Give examples.

Ans: 2's complement is a method used to represent **negative binary numbers** in computers.

◆ **How it works:**

To find the 2's complement of a binary number:

1. Invert all bits (0 becomes 1, and 1 becomes 0).
2. Add 1 to the result.

◆ **Why it's used:**

- Makes subtraction easy using addition.
- Helps the CPU process both positive and negative values.

◆ **Example:**

Find 2's complement of 5 (0101) in 4-bit binary:

1. Invert bits → 1010
2. Add 1 → $1010 + 0001 = 1011$ → This is -5 in 2's complement.

3. Define the terms Overflow and Underflow. Explain Overflow in number system with one example.

Ans:

◆ **Overflow:**

Occurs when the result of a calculation is **larger than the maximum value** the system can store.

◆ **Underflow:**

Occurs when the result is **too small (close to zero)** for the system to represent.

◆ **Overflow Example:**

In 4-bit binary, maximum positive number is 0111 (7).

If you add 0100 (4) + 0101 (5) = 1001 (9), it causes overflow as 9 cannot be represented in 4-bit signed binary.

4. What is the significance of ASCII and Unicode character coding schemes?

Ans: ASCII and Unicode are **character encoding schemes** that convert text into numbers so computers can understand it.

◆ **ASCII (American Standard Code for Information Interchange):**

- Uses **7 or 8 bits**.
- Can represent **128 or 256 characters** (mainly English).
- Example: A = 65, a = 97.

◆ **Unicode:**

- Uses **8 to 32 bits**.

- Can represent **all major languages** of the world.
- Example: A = U+0041, Urdu character "ا" = U+0627.

◆ **Significance:**

- ASCII is suitable for **basic English text**.
 - Unicode supports **globalization**, multilingual text, and emojis.
-

5. What is an operating system? Explain any five tasks/functions of OS.

Ans: An **Operating System (OS)** is software that acts as an **interface between the user and hardware**. It manages system resources and provides essential services.

◆ **Five functions of OS:**

1. **Process Management** – Controls running programs.
 2. **Memory Management** – Allocates memory to programs.
 3. **File Management** – Manages files and folders.
 4. **Device Management** – Controls hardware like printer, keyboard, etc.
 5. **User Interface** – Provides GUI or CLI to interact with the system.
-

6. Explain any four types of operating systems.

Ans:

1. **Batch Operating System:**
 - Processes jobs in batches with no interaction.
 - Example: Old IBM systems.
 2. **Time-Sharing Operating System:**
 - Allows multiple users to use the system at the same time.
 - Example: UNIX.
 3. **Real-Time Operating System (RTOS):**
 - Responds instantly to inputs, used in critical systems.
 - Example: Medical machines, autopilot.
 4. **Distributed Operating System:**
 - Controls multiple computers working together as one.
 - Example: LAN-based system sharing tasks.
-

7. How operating system manages applications? Explain five states of a process with diagram.

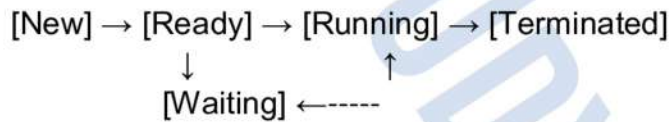
Ans:

The OS manages applications by **allocating CPU time, memory, and I/O devices**. It schedules processes, prevents conflicts, and ensures smooth execution.

◆ **Five Process States:**

1. **New** – Process is being created.
2. **Ready** – Waiting to be assigned to the CPU.
3. **Running** – Instructions are being executed.
4. **Waiting** – Waiting for I/O operations.
5. **Terminated** – Process has finished execution.

📝 *You may draw the diagram like this in exams:*



8. Explain the role of Integrated Development Environments (IDEs) in the software development process, including specific features that enhance programming efficiency.

Ans:

An **IDE** is a software suite that provides **tools for writing, testing, and debugging code**.

◆ **Key Features:**

- **Code Editor:** With syntax highlighting.
- **Compiler/Interpreter:** To convert code to machine language.
- **Debugger:** To find and fix errors.
- **Auto-complete:** Suggests code while typing.
- **Version Control:** Tracks code changes.

◆ **Role in Development:**

IDEs **increase coding speed**, reduce errors, and help beginners and professionals build software efficiently.

9. What is software hosting? Explain On-Premises and Cloud hosting. Also give two pros and cons of each.

Ans:

Software Hosting refers to **running and storing applications** on servers.

◆ **On-Premises Hosting:**

Software is hosted on **organization's own servers**.

✓ **Pros:**

1. Full control over data
2. High security (if well managed)

✗ **Cons:**

3. Expensive hardware setup
 4. Limited scalability
-

◆ **Cloud Hosting:**

Software is hosted on **remote servers via the internet**.

✓ **Pros:**

1. Cost-effective and scalable
2. Accessible from anywhere

✗ **Cons:**

3. Depends on internet connection
 4. Less control over data
-

The End

✓ **Subscribe for Learning**