INFORMATION TECHNOLOGY FOR STATISTICS

1. Role of Information Technology (IT) and Computers in Today's Society

Information Technology refers to the use of computers, software, networks, and digital systems to process, store, manage, and communicate information. IT has become central to all modern sectors.

Major Roles of IT

a) Education

- Supports teaching and learning through digital resources.
- Students use computers for research, assignments, and assessments.
- E-learning platforms enable remote education.

b) Business

- Automates operations and improves productivity.
- Supports decision-making through data analytics and management systems.
- Enables online transactions and global communication.

c) Industry

- Used in product design, production planning, manufacturing, and logistics.
- Enables robotics, automation, and inventory management systems.

d) Government

- Digital census systems for population statistics.
- Online taxation systems for filing and payments.
- Defense systems (e.g., missile trajectory calculations).

e) Healthcare

- Electronic medical records, patient scheduling, and billing.
- Medical imaging, laboratory systems, and telemedicine.

f) Marketing

- Enables digital advertising, social media marketing, and online product display.

g) Entertainment

- Streaming platforms, gaming, music, and digital media creation.

In Conclusion: IT enhances efficiency, accuracy, communication, and global connectivity in every sector.

2. Fundamentals of Computer Operations

Computer operations describe how a computer inputs, processes, stores, and outputs information.

a) Input

- The stage where raw data enters the computer.
- Devices: keyboard, mouse, camera, sensors, scanners, network interfaces.

b) Processing

- The CPU manipulates, calculates, and transforms input data based on instructions.
- Key components:
    - **CPU**: Executes instructions and controls system operations.
    - **ALU**: Performs arithmetic and logical comparisons.
    - **Memory (RAM)**: Temporarily stores instructions and data needed by the CPU.

c) Output

- Displays or sends out processed information.
- Devices: monitors, printers, speakers, network ports.

d) Storage

- Retains data permanently for future use.

Examples:

- **Primary storage**: RAM (volatile, temporary).
- **Secondary storage**: HDDs, SSDs, optical disks (non-volatile, permanent).

e) Control Unit

- Directs and coordinates the sequence of operations.
- Manages data flow between CPU, memory, and input/output devices.

Flow of Operation:
Input → Processing (CPU/ALU) → Storage (temporary/permanent) → Output

3. Basics of Computer Hardware and Software

Hardware

Physical components of a computer system.

Key Components

- **CPU**: Executes instructions; key specs include clock speed, core count, and cache.
- **Memory (RAM)**: Temporary, high-speed working area for active programs.
- **Storage Devices**:
    - **HDD** – mechanical, large capacity, slower.
    - **SSD** – flash memory, no moving parts, faster, more durable.
    - **NVMe SSD** – very high-speed SSDs using NVMe protocol.
- **Motherboard**: Connects all components; contains sockets, slots, and ports.
- **Power Supply Unit (PSU)**: Converts AC to DC power for internal components.
- **Input Devices**: Keyboard, mouse, scanners, sensors.
- **Output Devices**: Monitor, printers, speakers.
- **Expansion Cards**: Graphics card, sound card, network card.

Software

Programs or instructions that tell hardware what to do.

Types of Software

- **System Software**
    - **Operating System (OS)**: Manages hardware, files, memory, and apps.
    - **Device Drivers**: Allow OS to communicate with specific hardware.
    - **Utilities**: Backup, antivirus, file management tools.
- **Application Software**

    - Productivity tools: Word processors, spreadsheets.
    - Creative tools: Image/video editing, design software.
    - Browsers, communication apps, games.
- **Firmware**
    - Low-level software embedded in hardware devices for control and initialization.

Relationship between the two:
Hardware executes → Software instructs → Both rely on each other.

4. <u>Data Files</u>

Data files store information in structured or unstructured formats suitable for software processing.

a) CSV (Comma-Separated Values)

- Plain text files with values separated by commas.
- Ideal for tabular datasets.
  e.g. Name, age, course, grade (Jeff, 25, Mechanical Engineering, B)
- **Structure:**
    - Line 1: column headers
    - Lines below: data records
    - Use of commas to separate values. If a value contains a comma, enclose it in quotes.

  e.g. “Jeff, Paul”, 25, Mechanical Engineering, B
- Save with **.csv**

b) JSON (JavaScript Object Notation)

- Stores data as key–value pairs.
- Suitable for hierarchical or structured data.
- Structure: Uses arrays {}, values within arrays are separated by commas, and double quotes around strings.

e.g.

```
[
  {
    "Name": "Jeff Paul",
    "Age": "25",
    "Course": "Mechanical Engineering",
    "Grade": "A"
  },
]
```

- Save with **.json**

c) Plain Text Files (TXT)

- Unstructured simple text. (simple line by line text)
- Each record written in human-readable format.
- One record per line separated by commas.

  e.g.

  Name: Grace Hellen, Age: 19, Course: Financial Engineering
- Save with **.txt**

Best Practices for Creating Data Files

- Maintain consistent structure and formatting.
- Use correct delimiters.
- Validate JSON/XML syntax before use.
- Store using UTF-8 for international compatibility.
- Include headers to describe fields.

5. Means of Disk Storage

Disk storage technologies store digital data on physical disks, platters, or optical surfaces.

a) Hard Disk Drive (HDD)

- Uses **spinning magnetic platters** and mechanical read/write heads.
- Data stored magnetically in concentric tracks and sectors.

Characteristics

- Capacity: 500 GB – 16 TB
- Speed: Slower (mechanical components)
- Cost: Cheaper per GB
- Durability: Vulnerable to shock and mechanical wear
- Best suited for: Bulk storage, backups, archives

b) Solid State Drive (SSD)

- Uses **flash memory chips**, no moving parts.

Characteristics

- Capacity: 128 GB – 8 TB
- Speed: Extremely fast (quick boot & load times)
- Cost: Higher per GB
- Durability: Resistant to shock
- Ideal for: Operating systems, active software, and performance tasks

c) Hybrid Drive (SSHD)

- Combines HDD storage with SSD caching.
- Frequently used data stored on SSD part for faster access.

Characteristics

- Performance: Faster than HDD, slower than full SSD
- Cost: In-between SSD and HDD
- Good for: Laptop upgrades, mid-range systems

d) Optical Disks (CD, DVD, Blu-ray)

- Store data using **laser technology** that burns pits and lands on disk surfaces.

Characteristics

- Capacity:
    - CD: 700 MB
    - DVD: 4.7–9 GB

    - Blu-ray: 25–50 GB
- Low cost, slow read/write speed
- Prone to scratches
- Used for: Media distribution, software installation, long-term archives

e) External/Portable Drives

- External HDDs or SSDs in portable enclosures.
- Connected via USB, USB-C, or Thunderbolt.

Characteristics

- Portable, reliable for backups and data transfer
- Performance depends on drive type (HDD vs SSD)

6. Use the different number systems and bases and apply concepts in the 4 basic operations.

→ Number systems provide the mathematical language through which computers represent and process data.
→ In computing, various bases that are commonly used i.e binary, octal, decimal and hexadecimal are used to encode numbers efficiently depending on context. Understanding how to interpret, convert and manipulate numbers in these systems is essential for foundational computer science and digital electronics.

Examples:

1. Binary (Base 2)

→ Binary uses digits 0 and 1
→ Positional values are powers of 2.
→ It is essential to computing as it represents bits.
→ Used internally by all digital systems because it maps directly to electrical states (ON/OFF)

✓ Addition:

→ Example: Add 1011_2 (11 in decimal) and 110_2 (6 in decimal)

```
  1 1
  1 0 1 1
+   1 1 0
-------------
1 0 0 0 1
```

$= 10001_2$ (17 in decimal)

✓ Subtraction:

→ Example: Subtract 110_2 from 1011_2

```
  0 2
  1 ø 1 1
-   1 1 0
-------------
    1 0 1
```

$= 101_2$ (5 in decimal)

✓ Multiplication:

→ Example: $101_2 * 11_2$

```
    1 0 1
      1 1
-------------
    1 0 1
+ 1 0 1 0
-------------
1 1 1 1
```

$= 1111$ (15 in decimal)

✓ Division :

→ Example : $1101_2 \div 11_2$

```
      1 0 0
   ┌─────────
11 | 1 1 0 1
     1 1
   ─────
     0 0
       1 1
     ─────
     0 0 1
       1 1
     ─────
     0 0 1
```

$= 100_2$ remander 1
or
4 remainder 1

2. Octal (Base 8)

→ Uses digits 0–7
→ Positional values are powers of 8.
→ Used historically for compact binary representation.

✓ Addition : (Carry when sum ≥ 8)

→ Example : Add 27_8 and 14_8

```
  1
  2 7
+ 1 4
    11
   -8
  ----
  4 3
  ----
```

$= 43_8$ (35 in decimal)

✓ Subtraction : (Borrow 8 from left if top digit is smaller)

→ Example : $27_8 - 14_8$

```
  2 7
- 1 4
  ----
  1 3
  ----
```

$= 13_8$ (11 in decimal.)

✓ Multiplication : (Carry if result ≥ 8.)

→ Example : $12_8 \times 3_8$

```
  1 2
x   3
  ----
  3 6
  ----
```

$= 36_8$ (30 in decimal)

✓ Division :

→ Example: $27_8 \div 3_8$

```
  0 7
3|2 7
   ↓
  2 7
  2 5
  ---
  2
```

= 7 remainder 2

$(27)_8 = 2\times8^1 + 7\times8^0$

$16 + 7$

$= (23)_{10}$

$2\times8^1 + 5\times8^0 = (21)_{10}$

$\downarrow$

$(25)_8$

3. Decimal (Base 10)

→ This is our standard system (digits 0-9)
→ Positional values are powers of 10.
→ ~~Commonly used to represent Memory addresses and Machine code.~~
→ Primary system used in everyday arithmetic.

✓ Addition: (Carry when ≥10)

→ Example: $(23)_{10} + (12)_{10}$

```
 2 3
 1 2
 ---
 3 5
```

$= 35_{10}$

✓ Subtraction: (Borrow 10)

→ Example: $(23)_{10} - (12)_{10}$

```
 2 3
 1 2
 ---
 1 1
```

$= 11_{10}$

✓ Multiplication

→ Example: $(10)_{10} \times (3)_{10}$

```
 1 0
   3
 ---
 3 0
```

$= 30_{10}$

✓ Division :

→ Example: $(23)_{10} \div (3)_{10}$

$$
\begin{array}{r|l}
3 & 23 \\
 & 21 \\
\hline
 & 2
\end{array}
\qquad 7
$$

$= 7$ remainder 2

4. Hexadecimal (Base 16)

→ Hexadecimal uses digits 0-9 and A-F
→ Positional values are powers of 16.
→ Commonly used to represent Memory addresses and Machine code.

✓ Addition: (Carry when ≥ 16.)

→ Example: Add $2A_{16}$ and $1F_{16}$

$$
\begin{array}{r}
{\scriptstyle 1} \\
2\ A \\
+\ 1\ F \\
\hline
4\ 9
\end{array}
$$

A = 10
F = 15
10 + 15 = 25

25 - 16
= 9

$= 49_{16}$ (78 in decimal)

✓ Subtraction: (Borrow 16)

→ Example: $2A_{16} - 1F_{16}$

$$
\begin{array}{r}
{\scriptstyle 1} \\
2\ A^{(0+16)} \\
1\ F \\
\hline
0\ B
\end{array}
$$

A = 10
F = 15
26 - 15
= 11 (B)

$\Rightarrow B_{16} = 11$

✓ Multiplication :

→ Example : $2A_{16} - 1B_{16}$

A = 10
B = 11
10 x 11 = 110

110 ÷ 16 = 6 remainder 14
14 = E
$= 6E_{16}$

✓ Division :

→ Example : $2A_{16} \div B_{16}$

2A $\Rightarrow (2 \times 16) + 10$
$= 42$
B = 11

42 ÷ 11 = 3 remainder 9

$= (3 \text{ remainder } 9)_{16}$