



Kot Bhalwal, Jammu

Model Institute of Engineering
& Technology (Autonomous)
Dr. Arun K. Gupta Teaching-Learning Centre

Department of CSE
Details of Lesson Plan

S.No.	Particulars	Details
1.	Course Name	Computer Organization and Architecture.
2.	Course Code	COM-301
3.	Academic Year	2024-2025
4.	Semester	3 rd
5.	Number of Lesson plans	38
6.	Faculty Assigned	Harashleen kour

Faculty Signature



Version 1.1

श्रेष्ठ

श्रम

नवीनता

Please Do Not Print Unless Necessary

Lesson Plan No. 1	Course Name: Computer Organization and Architecture Topic: Overview of Digital Electronics	Course No.: COM-301
--------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of digital electronics. Identify different types of digital logic gates and their functions. Explain the significance of binary numbers in digital electronics. Recognize the applications of digital fundamentals in modern technology.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> What do you understand by the term "digital electronics"? How often do you encounter digital devices in your daily life? Explain the relevance of digital fundamentals in today's technology-driven world. Development (30 minutes) <ol style="list-style-type: none"> What are Digital Fundamentals? <ul style="list-style-type: none"> -Definition: Explain digital fundamentals as the principles underlying digital systems, where information is represented in binary form (0s and 1s). Emphasize their role in computers, mobile phones, and other digital devices. -Importance: Discuss how digital fundamentals enable the operation and functionality of various electronic devices. Highlight the importance of digital systems in data processing, storage, and communication. Basic Components of Digital Fundamentals: <ul style="list-style-type: none"> -Binary Numbers: The foundation of digital systems, where data is represented using two symbols (0 and 1). -Logic Gates: The building blocks of digital circuits that perform logical operations. -AND Gate: Outputs true only if all inputs are true. -OR Gate: Outputs true if at least one input is true. -NOT Gate: Inverts the input signal. -NAND, NOR, XOR, XNOR Gates: Variations of basic logic gates with specific functions. -Truth Tables: Tables that define the output of a logic gate for all possible input combinations. -Boolean Algebra: Mathematical notation used to describe logic gate functions and simplify digital circuits.

	<p>c. Applications of Digital Fundamentals:</p> <ul style="list-style-type: none"> • Real-World Applications <ul style="list-style-type: none"> -Discuss examples like digital clocks, calculators, and computer processors. -Highlight the use of digital logic in modern devices such as smartphones and smart home systems. <p>d. Case Study:</p> <ul style="list-style-type: none"> -Walk through the design of a simple digital circuit using basic logic gates. <p>e. YouTube Video:</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the basic components of digital fundamentals and their functions. - Discuss the answers with the class.
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading <ul style="list-style-type: none"> - "Digital Design" by M. Morris Mano, Chapter 1, pp. 1-35. - "Fundamentals of Digital Logic with Verilog Design" by Stephen Brown and Zvonko Vranesic, Chapter 2, pp. 40-65. 3. Homework <ul style="list-style-type: none"> - Write a short essay on the importance of binary numbers in digital electronics and upload it on the learning Camu.
Evaluation	<ol style="list-style-type: none"> 1. How would you define digital fundamentals in your own words? 2. What are the basic components of digital electronics? 3. Why are binary numbers essential in digital systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 2	Course Name: Computer Organization and Architecture Topic: Basics of Computer Organization and Architecture	Course No.: COM-301
--------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of computer organization and architecture. Identify the key components of a computer system and their functions. Explain the significance of different computer architectures. Recognize the applications of computer organization and architecture in modern computing.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> What do you understand by the term "computer organization and architecture"? How often do you think about the internal components when using a computer? Explain the relevance of understanding computer architecture in today's technology-driven world. <p>Development (30 minutes)</p> <ol style="list-style-type: none"> Computer Organization and Architecture: Definition <ul style="list-style-type: none"> Explain computer organization as the operational units and their interconnections that realize the architectural specifications. Explain computer architecture as the attributes of a system visible to a programmer, such as instruction sets, data formats, and addressing modes. Importance of Computer Organization and Architecture <ul style="list-style-type: none"> Discuss how these concepts enable the design and functionality of computers. Highlight their role in the performance, efficiency, and capabilities of computer systems. Basic Components of Computer Systems: <ul style="list-style-type: none"> Central Processing Unit (CPU): The brain of the computer that executes instructions. Memory: <ul style="list-style-type: none"> Primary Memory (RAM): Temporary storage for currently used data and instructions. Secondary Memory: Long-term storage for data and programs.



	<p>-Input/Output Devices (I/O): Interfaces for the computer to interact with the external environment (e.g., keyboard, mouse, monitor).</p> <p>-System Bus: The communication pathway connecting various components of the computer.</p> <p>d. Computer Architectures:</p> <ul style="list-style-type: none"> • Von Neumann Architecture: -Discuss its characteristics such as shared program and data memory. • Harvard Architecture: -Explain its separate storage and signal pathways for - instructions and data. <p>e. Real-World Applications: -Discuss examples such as personal computers, smartphones, and servers. -Highlight the use of different architectures in various devices.</p> <p>f. Case Study: -Walk through the design of a simple computer system using the Von Neumann architecture.</p> <p>g. YouTube Video:</p> <p>2. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the basic components of computer systems and their functions. - Discuss the answers with the class.
<p>Closure</p>	<p>1. Closure</p> <ul style="list-style-type: none"> - Summarize the Lesson Learning Outcomes and get affirmation from students on these. <p>2. Suggested Reading</p> <ul style="list-style-type: none"> - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 1, pp. 1-35. - "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 2, pp. 40-65
<p>Evaluation</p>	<ol style="list-style-type: none"> 1. How would you define computer organization and architecture in your own words? 2. What are the basic components of a computer system? 3. Why is it important to understand different computer architectures? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>



Lesson Plan No. 3	Course Name: Computer Organization and Architecture Topic: Conversions	Course No.: COM-301
--------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the fundamental concepts of number system conversions. Convert numbers between different bases (binary, decimal, octal, and hexadecimal). Explain the significance of various number systems in computer architecture. Apply conversion techniques in practical computing scenarios.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask students: "What number systems are you familiar with?" Discuss the importance of different number systems in computing. Explain the relevance of understanding conversions in computer architecture. Development (30 minutes) <ol style="list-style-type: none"> Number Systems Overview <ul style="list-style-type: none"> -Definition: Explain the concept of number systems and their bases (e.g., binary is base-2, decimal is base-10). -Importance: Discuss how computers use different number systems for various operations. Binary System (Base-2) <ul style="list-style-type: none"> -Conversion to Decimal: Explain the method of converting binary numbers to decimal. -Example: Convert 110121101_211012 to decimal. -Conversion from Decimal: Explain the method of converting decimal numbers to binary. Example: Convert 131013_{10}1310 to binary. Octal System (Base-8) <ul style="list-style-type: none"> -Conversion to Decimal: Explain the method of converting octal numbers to decimal. -Example: Convert 1578157_81578 to decimal. -Conversion from Decimal: Explain the method of converting decimal numbers to octal. -Example: Convert 11110111_{10}11110 to octal. Hexadecimal System (Base-16) <ul style="list-style-type: none"> -Conversion to Decimal: Explain the method of converting hexadecimal numbers to decimal. -Example: Convert 1A3161A3_{16}1A316 to decimal. -Conversion from Decimal: Explain the method of converting decimal numbers to hexadecimal.



	<p>-Example: Convert 41910419_{10} to hexadecimal.</p> <p>e. Inter-conversion between Non-decimal Bases</p> <p>-Explain methods for converting directly between binary, octal, and hexadecimal.</p> <p>-Example: Convert 1101012110101_2 to octal and hexadecimal.</p> <p>f. Applications in Computing</p> <p>-Discuss the role of different number systems in computer memory addressing, instruction sets, and data representation.</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none">● Ask students to perform a few conversion exercises:<ul style="list-style-type: none">-Convert 101021010_2 to decimal.-Convert 25510255_{10} to hexadecimal.-Convert 3778377_8 to binary.
Closure	<ol style="list-style-type: none">1. Summarize the key points of number system conversions.2. Ensure students can explain the conversion processes and their importance in computing.3. Suggested Reading<ul style="list-style-type: none">- "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 60-85.- "Digital Logic and Computer Design" by M. Morris Mano, Chapter 1, pp. 10-30.
Evaluation	<ol style="list-style-type: none">1. How would you define number system conversions?2. Convert 101021010_2 to decimal.3. Why is it important to understand different number systems in computer architecture? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>



Lesson Plan No. 4	Course Name: Computer Organization and Architecture Topic: Logic Gates	Course No.: COM-301
--------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of logic gates. Identify different types of logic gates and their functions. Explain the significance of logic gates in digital electronics. Recognize the applications of logic gates in modern technology. 																								
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video 																								
Teaching Development	<p>1. Introduction (5 minutes)</p> <ul style="list-style-type: none"> - What do you understand by the term "logic gates"? - How often do you encounter digital devices that use logic gates in your daily life? - Introduce logic gates as the fundamental building blocks of digital circuits. - Explain the relevance of logic gates in today's technology-driven world. <p>2. Development (30 minutes)</p> <p>a. What are Logic Gates?</p> <ul style="list-style-type: none"> -Definition: Define logic gates as electronic devices that perform logical operations on one or more binary inputs to produce a single binary output. -Importance: Discuss how logic gates enable the operation and functionality of various digital circuits and systems. <p>b. Types of Logic Gates:</p> <ul style="list-style-type: none"> -AND Gate: Outputs true (1) only if all inputs are true (1). -Truth Table: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>A</th> <th>B</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> </tr> </tbody> </table> <ul style="list-style-type: none"> -OR Gate: Outputs true (1) if at least one input is true (1). -Truth Table: <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>A</th> <th>B</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	A	B	Output	0	0	0	0	1	0	1	0	0	1	1	1	A	B	Output	0	0	0	0	1	1
A	B	Output																							
0	0	0																							
0	1	0																							
1	0	0																							
1	1	1																							
A	B	Output																							
0	0	0																							
0	1	1																							

1	0	1
1	1	1

-NOT Gate: Inverts the input signal.

-Truth Table:

A	Output
0	1
1	0

-NAND Gate: Outputs false (0) only if all inputs are true (1).

-Truth Table:

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

-NOR Gate: Outputs true (1) only if all inputs are false (0).

-Truth Table:

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

-XOR Gate: Outputs true (1) if inputs are different.

-Truth Table:

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

-XNOR Gate: Outputs true (1) if inputs are the same.

-Truth Table:

	A	B	Output
	0	0	1
	0	1	0
	1	0	0
	1	1	1

c. Boolean Algebra:
 -Explanation: Introduce Boolean algebra as a mathematical notation used to describe logic gate functions and simplify digital circuits.
 -Examples: Demonstrate simple Boolean expressions and their corresponding logic gate implementations.

d. Applications of Logic Gates:
 -Real-World Applications: Discuss examples like digital clocks, calculators, and computer processors.
 -Highlight the use of logic gates in modern devices such as smartphones and smart home systems.

e. Case Study:
 -Walk through the design of a simple digital circuit using basic logic gates.

f. YouTube Video:

3. Exercise (5 minutes)

- Activity:**
 - Ask students to summarize the basic components of logic gates and their functions.
 - Discuss the answers with the class.

Closure

- Recap the key points covered in the lesson.
- Get affirmation from students on their understanding of the lesson.
- Suggested Reading
 - "Digital Design" by M. Morris Mano, Chapter 1, pp. 1-35.
 - "Fundamentals of Digital Logic with Verilog Design" by Stephen Brown and Zvonko Vranesic, Chapter 2, pp. 40-65.

Evaluation

- How would you define logic gates in your own words?
- What are the basic types of logic gates?
- Why are logic gates essential in digital systems?

Spend 5 minutes to wrap up and consolidate the learnings.



Lesson Plan No. 5	Course Name: Computer Organization and Architecture Topic: Logic Gates	Course No.: COM-301
--------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of complements in COA. Differentiate between various types of complements. Apply the concept of complements in arithmetic operations.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) Discussion Questions: <ul style="list-style-type: none"> -What do you understand by the term "complement" in mathematics? -Can you recall any scenarios where you have used complements? -Explain the importance of complements in computer arithmetic, particularly in simplifying the subtraction operation and representing negative numbers. Development (30 minutes) <ul style="list-style-type: none"> • Definition: <ul style="list-style-type: none"> -Explain complements as methods to transform numbers to facilitate arithmetic operations in digital systems. • Types of Complements: <ul style="list-style-type: none"> 1's Complement: <ul style="list-style-type: none"> -Definition: The 1's complement of a binary number is obtained by inverting all bits (changing 0s to 1s and 1s to 0s). -Example: For binary number 1010, the 1's complement is 0101. 2's Complement: <ul style="list-style-type: none"> -Definition: The 2's complement of a binary number is obtained by adding 1 to the 1's complement of the number. -Example: For binary number 1010, the 1's complement is 0101, and the 2's complement is 0110. • Applications of Complements: <ul style="list-style-type: none"> -Arithmetic Operations: Simplifying subtraction by converting it to addition using 2's complement. -Representing negative numbers in binary systems. -Problem Solving: Example problems demonstrating subtraction using 2's complement. -Illustrate how complements make it easier to perform binary arithmetic operations in digital systems. • YouTube Video: Exercise (5 minutes)

	<ul style="list-style-type: none"> - Ask students to compute the 1's and 2's complement of a given binary number. - Discuss the solutions and ensure understanding.
Closure	<ol style="list-style-type: none"> 1. Recap the concepts of 1's and 2's complements, their calculation, and applications. 2. Suggested Reading <ul style="list-style-type: none"> - "Computer System Architecture" by M. Morris Mano, Chapter 3, pp. 50-75. - "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 2, pp. 30-60.
Evaluation	<ol style="list-style-type: none"> 1. How would you define 1's and 2's complement in your own words? 2. What are the steps to calculate the 2's complement of a binary number? 3. Why are complements essential in computer arithmetic? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 6	Course Name: Computer Organization and architecture Topic: Register Transfer Language	Course No.: COM-301
--------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of Register Transfer Language (RTL). Identify and describe the different types of registers used in RTL. Explain the significance of micro-operations and how they are represented in RTL. Recognize the applications of RTL in the design and functioning of digital systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Begin by asking: "What do you understand by the term 'Register Transfer Language'?" Discuss how RTL is encountered in computer architecture and its importance in digital system design. Explain the relevance of RTL in understanding the operations within a computer processor. Development (30 minutes) <ol style="list-style-type: none"> What is Register Transfer Language? <ul style="list-style-type: none"> Definition: <ul style="list-style-type: none"> -Explain RTL as a language used to describe the operations, data flow, and data transfer between the registers within a digital system. Highlight the use of symbolic notation to represent micro-operations and data transfers. Basic Components of Register Transfer Language: <ul style="list-style-type: none"> Registers: <ul style="list-style-type: none"> -Discuss the role of registers in a digital system as small storage locations that hold data temporarily. Explain different types of registers such as accumulator, data register, address register, and status register. Micro-operations: <ul style="list-style-type: none"> -Define micro-operations as the basic operations performed on the data stored in registers. Examples include data transfer operations, arithmetic operations, logical operations, and shift operations. -Provide symbolic representations for common micro-operations, such as:



	<p>-Data transfer: $R2 \leftarrow R1$ (Transfer data from register R1 to register R2) -Addition: $R3 \leftarrow R1 + R2$ (Add the contents of registers R1 and R2 and store the result in R3)</p> <p>c. Importance of Register Transfer Language: -Discuss how RTL helps in designing and understanding the internal functioning of digital systems, especially in the development of microprocessors and control units. -Highlight the role of RTL in simplifying the design process by providing a clear and concise way to represent complex operations.</p> <p>d. Applications of Register Transfer Language:</p> <ul style="list-style-type: none"> ● Real-World Applications: -Discuss examples like the design of arithmetic logic units (ALUs), control units, and data paths in microprocessors. -Highlight the use of RTL in hardware description languages (HDLs) such as VHDL and Verilog for designing and simulating digital circuits. ● Case Study: -Walk through the design of a simple arithmetic unit using RTL notation. Explain each step and the corresponding RTL representation. ● YouTube Video: <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the basic components of RTL and their functions. - Discuss the answers with the class to ensure understanding.
<p>Closure</p>	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading: <ul style="list-style-type: none"> - "Computer System Architecture" by M. Morris Mano, Chapter 4, pp. 130-175. - "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 5, pp. 200-250.
<p>Evaluation</p>	<ol style="list-style-type: none"> 1. How would you define Register Transfer Language in your own words? 2. What are the basic components of RTL? 3. Why are micro-operations essential in digital systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 7	Course Name: Computer Organization and architecture Topic: Register transfer	Course No.: COM-301
--------------------------	---	----------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of register transfer in computer architecture. Identify different types of registers and their functions. Explain the significance of register transfer in the execution of instructions. Recognize the applications of register transfer in computer systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> What do you understand by the term "register transfer"? How often do you think registers are used in a computer's operation? Explain the relevance of register transfer in the functioning of computer systems. Development (30 minutes) <ol style="list-style-type: none"> What is Register Transfer? <ul style="list-style-type: none"> Definition: <p>-Explain register transfer as the process of transferring data between registers, which are small, fast storage locations within a CPU. Emphasize their role in executing instructions.</p> Importance: <p>-Discuss how register transfers are fundamental to the CPU's operation, enabling data processing, arithmetic operations, and instruction execution.</p> Basic Components of Register Transfer: <ul style="list-style-type: none"> -Registers: Define registers and explain their purpose as temporary storage locations for data and instructions. -Register Transfer Language (RTL): Explain RTL as a symbolic notation used to describe the operations, data flow, and control signals involved in register transfers. -Micro-operations: Describe micro-operations as the fundamental operations performed on data stored in registers, such as data transfer, arithmetic operations, and logical operations.

	<p>-Control Signals: Discuss the role of control signals in coordinating the movement of data between registers and the execution of micro-operations.</p> <p>c. Types of Registers:</p> <p>-General-Purpose Registers: Explain their use in storing temporary data and intermediate results during instruction execution.</p> <p>-Special-Purpose Registers: Introduce examples like the Program Counter (PC), Instruction Register (IR), and Memory Address Register (MAR), explaining their specific functions.</p> <p>d. Applications of Register Transfer:</p> <p>-Real-World Applications: Discuss examples such as CPU operations, data processing tasks, and the functioning of modern computing devices.</p> <p>-Case Study: Walk through a simple example of register transfer in a basic CPU instruction cycle, explaining each step and the registers involved.</p> <p>e. YouTube Video:</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the basic components and types of registers involved in register transfer. - Discuss the answers with the class.
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson's learning outcomes and get affirmation from students on these. 2. Suggested Reading: <ul style="list-style-type: none"> - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 150-190. - "Computer System Architecture" by M. Morris Mano, Chapter 5, pp. 200-245.
Evaluation	<ol style="list-style-type: none"> 1. How would you define register transfer in your own words? 2. What are the basic components involved in register transfer? 3. Why are registers essential in computer systems?

Lesson Plan No. 8	Course Name: Computer Organization and architecture Topic: Bus and Memory Transfer	Course No.: COM-301
--------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of a bus in computer architecture. Describe the types of buses and their functions. Explain memory transfer processes and mechanisms. Identify the significance of bus and memory transfer in computer systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> What do you understand by the term "bus" in computer architecture? How do you think data is transferred between different components of a computer system? Explain the importance of buses and memory transfer in computer performance and efficiency. Development (30 minutes) <ol style="list-style-type: none"> What is a Bus? <ul style="list-style-type: none"> -Definition: A bus is a communication system that transfers data between components inside a computer or between computers. It consists of multiple wires or lines to carry signals and data. -Importance: Buses are critical for the operation of a computer system as they facilitate communication between the CPU, memory, and peripheral devices, enabling data transfer and synchronization. Types of Buses: <ul style="list-style-type: none"> -Data Bus: Transfers actual data between the processor, memory, and peripheral devices. -Address Bus: Carries the address location of where the data is to be read or written. It is unidirectional. -Control Bus: Transfers control signals from the CPU to other components, coordinating data transfers and ensuring proper communication. Memory Transfer: <ul style="list-style-type: none"> -Definition: Memory transfer refers to the process of moving data between the main memory and the processor or between different memory locations. -Mechanisms:

	<p>-Direct Memory Access (DMA): Allows peripheral devices to directly transfer data to/from memory without CPU intervention, increasing efficiency.</p> <p>-Memory-Mapped I/O: Peripheral devices are assigned specific memory addresses, enabling the CPU to read/write data as if it were memory.</p> <p>d. Importance of Bus and Memory Transfer:</p> <p>-System Performance: Efficient bus and memory transfer mechanisms are essential for high-performance computing, as they reduce bottlenecks and ensure smooth data flow.</p> <p>-Real-World Applications: Examples include data transfer in gaming consoles, real-time data processing in medical equipment, and high-speed data communication in networking devices.</p> <p>e. YouTube Video:</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the different types of buses and their roles in memory transfer. Discuss the answers with the class.
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and ensure students understand the key concepts. 2. Suggested Reading: <ul style="list-style-type: none"> - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 5, pp. 200-235. - "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 3, pp. 100-145.
Evaluation	<ol style="list-style-type: none"> 1. How would you define a bus in computer architecture in your own words? 2. What are the different types of buses and their functions? 3. Why is memory transfer essential in computer systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>



Kot Bhalwal, Jammu



Lesson Plan No. 9	Course Name: Computer Organization and architecture Topic: Arithmetic Micro-Operations	Course No.: COM-301
--------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: a. Understand the concept and significance of arithmetic micro-operations in computer architecture. b. Identify and describe different types of arithmetic micro-operations. c. Explain how arithmetic micro-operations are implemented in digital systems. d. Apply arithmetic micro-operations in practical scenarios.
Teaching Aids (if any)	a. Presentation slides. b. Projector c. You Tube video
Teaching Development	1. Introduction (5 minutes) <ul style="list-style-type: none">- What do you understand by the term "arithmetic micro-operations"?- Can you give examples of basic arithmetic operations you use in daily calculations?- Explain the relevance of arithmetic micro-operations in computer systems. 2. Development (30 minutes) <p>a. What are Arithmetic Micro-Operations? -Definition: Explain arithmetic micro-operations as elementary operations performed on binary data stored in registers within a computer system. These operations are essential for executing arithmetic functions in CPUs. -Importance: Discuss how arithmetic micro-operations facilitate various computational tasks, including addition, subtraction, increment, decrement, and shift operations, enabling complex arithmetic calculations in processors.</p> <p>b. Types of Arithmetic Micro-Operations: -Addition: Binary addition of two register contents, generating a sum and carry. -Subtraction: Achieved by adding the complement of the subtrahend to the minuend. -Increment: Increasing the binary value of a register by one. -Decrement: Decreasing the binary value of a register by one. Arithmetic Shift Left/Right: Shifting the bits of a register to the left or right, affecting the binary value and potentially multiplying or dividing by two.</p>

	<p>c. Implementation in Digital Systems:</p> <ul style="list-style-type: none"> -Binary Adder-Subtractor: A circuit that can perform both addition and subtraction based on control signals. -Arithmetic Logic Unit (ALU): The component of the CPU where arithmetic micro-operations are executed. <p>d. Applications of Arithmetic Micro-Operations:</p> <ul style="list-style-type: none"> -Real-World Applications: Discuss examples such as mathematical computations in processors, data manipulation in memory, and algorithms in various software applications. -Case Study: Walk through the design and implementation of a simple arithmetic circuit using binary adder and subtractor components. <p>e. YouTube Video:</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the types of arithmetic micro-operations and their functions. - Discuss the answers with the class.
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading: <ul style="list-style-type: none"> - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 85-120. - "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 4, pp. 150-190.
Evaluation	<ol style="list-style-type: none"> 1. How would you define arithmetic micro-operations in your own words? 2. What are the different types of arithmetic micro-operations? 3. How are arithmetic micro-operations implemented in digital systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>



Lesson Plan No. 10	Course Name: Computer Networks Topic: Instruction Cycle	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concept of the instruction cycle. Identify the different phases of the instruction cycle. Explain the significance of each phase in the execution of instructions. Recognize the applications of the instruction cycle in modern computer systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> What do you understand by the term "instruction cycle"? How often do you think the instruction cycle is repeated in a computer's operation? Explain the relevance of understanding the instruction cycle in today's computing technology. <p>Development (30 minutes)</p> <p>What is an Instruction Cycle?</p> <ul style="list-style-type: none"> Definition: <ul style="list-style-type: none"> Explain the instruction cycle as the process by which a computer retrieves, decodes, and executes an instruction from memory. Emphasize its role in the functioning of a CPU. Importance: <ul style="list-style-type: none"> Discuss how the instruction cycle is fundamental to the operation and performance of a computer. Highlight the continuous and rapid execution of the cycle in processing data and running applications. <p>Phases of the Instruction Cycle:</p> <ul style="list-style-type: none"> Fetch Phase: <ul style="list-style-type: none"> Definition: <ul style="list-style-type: none"> The CPU fetches the instruction from memory. This phase involves reading the instruction stored at the address pointed to by the program counter (PC). Components: <ul style="list-style-type: none"> Program Counter (PC): Holds the address of the next instruction to be fetched. Memory Address Register (MAR): Holds the address of the memory location to be read. Memory Buffer Register (MBR): Temporarily holds the data read from memory.

- **Decode Phase:**
 - **Definition:**
 - The CPU decodes the fetched instruction to determine the operation to be performed and the operands to be used.
 - **Components:**
 - Instruction Register (IR): Holds the fetched instruction for decoding.
 - Control Unit: Interprets the instruction and generates the necessary control signals.
- **Execute Phase:**
 - **Definition:**
 - The CPU executes the decoded instruction. This may involve arithmetic operations, data transfer, or logical operations.
 - **Components:**
 - Arithmetic Logic Unit (ALU): Performs the arithmetic and logical operations.
 - Registers: Temporary storage for operands and results.
- **Store Phase:**
 - **Definition:**
 - The CPU stores the result of the executed instruction in memory or a register.
 - **Components:**
 - Memory: For storing data and results.
 - Registers: For holding data temporarily.

Applications of the Instruction Cycle:

- **Real-World Applications:**
 - Discuss examples like CPU processing in personal computers, smartphones, and embedded systems.
 - Highlight how the efficiency of the instruction cycle impacts the performance of computing devices.

Case Study:

- Walk through the execution of a simple instruction set, illustrating each phase of the instruction cycle using basic operations like addition or data transfer.

YouTube Video:

Exercise (5 minutes)

- Ask students to summarize the phases of the instruction cycle and their significance.
- Discuss the answers with the class.

Closure	<ul style="list-style-type: none"> Summarize the lesson learning outcomes and get affirmation from students on these. <p>Suggested Reading</p> <ul style="list-style-type: none"> "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 123-145. "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 3, pp. 87-112.
Evaluation	<ul style="list-style-type: none"> How would you define the instruction cycle in your own words? What are the basic phases of the instruction cycle? Why is the instruction cycle essential in computer systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 11	Course Name: Computer Organization and architecture Topic: Shift Micro Operation	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of shift micro operations. Identify different types of shift operations. Explain the significance of shift operations in computer architecture. Demonstrate how shift operations are used in various applications.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Begin with a brief discussion: "What do you understand by the term 'shift operation' in computer architecture?" Ask: "Can you think of any instances where shifting data might be useful?" Explain the relevance of shift operations in data processing and manipulation within computer systems. Development (30 minutes) <ol style="list-style-type: none"> What are Shift Micro Operations? <ul style="list-style-type: none"> -Definition: Shift micro operations are used to shift the bits of a binary number to the left or right. -These operations are fundamental in various arithmetic and logical operations in a computer system. -Importance: Shifting bits is crucial for multiplication and division operations. They help in aligning data and manipulating bit patterns for efficient computation. Types of Shift Operations: <ul style="list-style-type: none"> -Logical Shift: Shifts bits to the left or right, with zeros filling the vacated positions. Example: Logical left shift of 1010 becomes 0100. -Arithmetic Shift: Similar to logical shift but maintains the sign bit (for signed numbers). Example: Arithmetic right shift of 1101 becomes 1110. -Circular Shift (Rotate): Bits shifted out from one end are reintroduced at the other end. Example: Circular left shift of 1011 becomes 0111 (and the leftmost bit is reintroduced at the rightmost position). Applications of Shift Operations: Multiplication and Division: Shifting left by one position is equivalent to

	<p>multiplying by 2; shifting right by one position is equivalent to dividing by 2.</p> <ul style="list-style-type: none"> -Data Packing and Unpacking: Efficiently manage data storage by packing multiple smaller data units into a larger one or unpacking them. -Bitwise Operations: Used in various algorithms, encryption, and encoding techniques. <p>d. Practical Examples and Demonstration:</p> <ul style="list-style-type: none"> -Show examples of shift operations in binary numbers using slides. -Demonstrate a simple circuit design using basic shift registers. -Discuss a case study of a real-world application, such as data encryption, where shift operations are used. <p>e. YouTube Video:</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the types of shift operations and their specific use cases. - Discuss the answers with the class.
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and confirm students' understanding. 2. Suggested Reading: <ul style="list-style-type: none"> - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 200-240. - "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 5, pp. 150-180.
Evaluation	<ol style="list-style-type: none"> 1. How would you define shift micro operations in your own words? 2. What are the different types of shift operations and their significance? 3. Why are shift operations essential in computer systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 12	Course Name: Computer Networks Topic: Instruction Code	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of instruction codes. Identify different types of instruction formats. Explain the significance of opcode and operand in an instruction. Recognize the role of instruction codes in CPU operations.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> Start with a question: "What do you understand by the term 'instruction code'?" Discuss how instructions are essential for a computer to perform tasks. Briefly explain the relevance of instruction codes in the CPU's operation and overall system functionality. <p>Development (30 minutes)</p> <p>What are Instruction Codes?</p> <ul style="list-style-type: none"> Definition: Instruction codes are binary codes that specify operations to be performed by the CPU. They are a part of the instruction set architecture (ISA). Importance: Explain how instruction codes enable the CPU to perform various tasks like data manipulation, control operations, and input/output operations. <p>Basic Components of Instruction Codes:</p> <ul style="list-style-type: none"> Opcode (Operation Code): <ul style="list-style-type: none"> The part of the instruction that specifies the operation to be performed (e.g., add, subtract, load, store). Operand: <ul style="list-style-type: none"> The part of the instruction that specifies the data or the address of the data on which the operation is to be performed. <p>Instruction Formats:</p> <ul style="list-style-type: none"> Discuss different instruction formats like: <ul style="list-style-type: none"> Zero Address Instructions: (e.g., stack-based operations) One Address Instructions: (e.g., accumulator-based operations) Two Address Instructions: (e.g., direct memory operations) Three Address Instructions: (e.g., register operations)



	<p>Role of Instruction Codes in CPU Operations:</p> <ul style="list-style-type: none">• Explain how the CPU fetches, decodes, and executes instructions.• Discuss the instruction cycle and how instruction codes are crucial for its execution. <p>Examples and Case Studies:</p> <ul style="list-style-type: none">• Provide examples of simple instruction codes and how they are used in basic CPU operations.• Walk through a case study of a simple program and explain how the instruction codes are executed. <p>YouTube Video:</p> <p>Exercise (5 minutes)</p> <ul style="list-style-type: none">• Ask students to write down an example of an instruction code and explain its components.• Discuss the examples with the class.
Closure	<ul style="list-style-type: none">• Summarize the lesson by reiterating the key points about instruction codes, their components, and their importance in CPU operations.• Confirm that students understand the lesson's learning outcomes. <p>Suggested Reading</p> <ul style="list-style-type: none">• "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 45-85.• "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 2, pp. 60-90.
Evaluation	<ul style="list-style-type: none">• How would you define an instruction code in your own words?• What are the basic components of an instruction code?• Why are opcodes essential in the instruction cycle? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 13	Course Name: Computer Networks Topic: Computer Register	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of computer registers. Identify different types of registers and their functions. Explain the significance of registers in CPU operations. Recognize the applications of registers in modern computer systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> What do you understand by the term "computer register"? How do you think registers are used in the CPU? Explain the relevance of registers in performing CPU operations. <p>Development (30 minutes)</p> <p>What are Computer Registers?</p> <ul style="list-style-type: none"> Definition: <ul style="list-style-type: none"> Explain computer registers as small, fast storage locations within the CPU used to hold data and instructions temporarily. Importance: <ul style="list-style-type: none"> Discuss how registers enable quick data access and manipulation during CPU operations. Highlight their role in executing instructions efficiently. <p>Types of Computer Registers:</p> <ul style="list-style-type: none"> General-Purpose Registers: <ul style="list-style-type: none"> Used for general data storage and arithmetic operations. Examples include AX, BX, CX, and DX in x86 architecture. Special-Purpose Registers: <ul style="list-style-type: none"> Used for specific functions within the CPU. Examples include: <ul style="list-style-type: none"> Program Counter (PC): Holds the address of the next instruction to be executed. Instruction Register (IR): Holds the currently executing instruction. Accumulator (ACC): Used in arithmetic and logic operations. Stack Pointer (SP): Points to the top of the stack in memory. Status Register (FLAGS): Holds condition codes and control bits.



	<p>Functionality of Registers:</p> <ul style="list-style-type: none">● Data Transfer:<ul style="list-style-type: none">○ Registers facilitate data transfer between the CPU and memory or I/O devices.● Arithmetic and Logic Operations:<ul style="list-style-type: none">○ Registers store operands for arithmetic and logic operations performed by the ALU.● Control Operations:<ul style="list-style-type: none">○ Registers control the flow of instructions and manage execution states. <p>Applications of Registers in Modern Computer Systems:</p> <ul style="list-style-type: none">● Real-World Applications:<ul style="list-style-type: none">○ Discuss examples such as the use of registers in executing machine-level instructions in processors.○ Highlight the role of registers in handling interrupts and context switching in operating systems.● Case Study:<ul style="list-style-type: none">○ Walk through the use of registers in a simple assembly language program. <p>YouTube Video:</p> <p>Exercise (5 minutes)</p> <ul style="list-style-type: none">● Ask students to summarize the types of registers and their functions.● Discuss the answers with the class.
Closure	<ul style="list-style-type: none">● Summarize the lesson learning outcomes and get affirmation from students on these. <p>Suggested Reading</p> <ul style="list-style-type: none">● "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 85-120.● "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 2, pp. 50-80.
Evaluation	<ul style="list-style-type: none">● How would you define computer registers in your own words?● What are the basic types of registers in the CPU?● Why are registers essential in CPU operations? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 14	Course Name: Computer Networks Topic: Computer Instructions	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of computer instructions. Identify different types of computer instructions and their formats. Explain the significance of instruction sets in computer architecture.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> Start with a question: "What do you think happens when you give a command to a computer?" Discuss the basic idea that computers follow instructions to perform tasks. Explain the importance of understanding computer instructions in the field of computer science and engineering. <p>Development (30 minutes)</p> <p>What are Computer Instructions?</p> <ul style="list-style-type: none"> Definition: Explain that computer instructions are commands that tell the computer what operations to perform. These instructions are part of the instruction set of a computer architecture. Importance: Discuss how computer instructions are crucial for executing programs, controlling hardware, and processing data. <p>Basic Components of Computer Instructions:</p> <ul style="list-style-type: none"> Instruction Format: Explain the structure of an instruction, typically consisting of an opcode (operation code) and operands (data or memory locations). <ul style="list-style-type: none"> Opcode: The part of an instruction that specifies the operation to be performed (e.g., ADD, SUB, LOAD). Operands: The data on which the operation is performed or the addresses where data is stored. <p>Types of Computer Instructions:</p> <ul style="list-style-type: none"> Data Transfer Instructions: Instructions that move data between registers, memory, and I/O devices (e.g., MOV, LOAD, STORE). Arithmetic Instructions: Instructions that perform arithmetic operations (e.g., ADD, SUB, MUL, DIV). Logical Instructions: Instructions that perform logical operations (e.g., AND, OR, NOT).



	<ul style="list-style-type: none">● Control Instructions: Instructions that change the sequence of execution (e.g., JMP, CALL, RET).● Input/Output Instructions: Instructions that handle I/O operations (e.g., IN, OUT). <p>Real-World Applications:</p> <ul style="list-style-type: none">● Discuss how different types of instructions are used in various applications, such as operating systems, application software, and embedded systems.● Case Study: Walk through a simple program and identify the types of instructions used. <p>YouTube Video:</p> <p>Exercise (5 minutes)</p> <ul style="list-style-type: none">● Ask students to summarize the basic types of computer instructions and their functions.● Discuss the answers with the class.
Closure	Summarize the lesson learning outcomes and get affirmation from students on their understanding. Suggested Reading: <ul style="list-style-type: none">● "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 2, pp. 50-90.● "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 3, pp. 100-130.
Evaluation	<ul style="list-style-type: none">● How would you define computer instructions in your own words?● What are the basic components of a computer instruction?● Why are instruction sets essential in computer systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 15	Course Name: Computer Networks Topic: Timing and Control Signals	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of timing and control signals. Identify different types of timing and control signals used in computer architecture. Explain the significance of these signals in the execution of instructions.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> Question: What do you understand by the term "timing and control signals"? Discussion: How often do you think about the processes that occur within a computer when it executes an instruction? Relevance: Explain the importance of timing and control signals in ensuring that instructions are executed correctly and efficiently within a CPU. <p>Development (30 minutes)</p> <ol style="list-style-type: none"> What are Timing and Control Signals? <ul style="list-style-type: none"> Definition: <ul style="list-style-type: none"> Timing signals coordinate the timing of various operations within the CPU. Control signals manage the actions of the CPU, ensuring that the correct operations are performed at the right time. Importance: <ul style="list-style-type: none"> Discuss how these signals are crucial for synchronizing the operations of various components within the CPU. Highlight their role in the proper execution of instructions, data transfer, and control flow. Types of Timing and Control Signals: <ul style="list-style-type: none"> Clock Signal: <ul style="list-style-type: none"> The clock signal provides a consistent timing reference for the sequential operations of the CPU. Control Signals: <ul style="list-style-type: none"> Fetch: Signals the retrieval of an instruction from memory. Decode: Initiates the interpretation of the fetched instruction. Execute: Triggers the execution of the instruction. Memory Read/Write: Manages the reading from and writing to memory.

	<ul style="list-style-type: none"> ■ Interrupts: Signals that an external event requires immediate attention from the CPU. <p>3. Significance of Timing and Control Signals:</p> <ul style="list-style-type: none"> ○ Synchronization: <ul style="list-style-type: none"> ■ Ensures all parts of the CPU work together harmoniously. ○ Efficiency: <ul style="list-style-type: none"> ■ Minimizes delays and maximizes throughput by coordinating operations. ○ Reliability: <ul style="list-style-type: none"> ■ Prevents errors by providing clear instructions for each operation. <p>4. Applications in Modern Computer Systems:</p> <ul style="list-style-type: none"> ○ Real-World Examples: <ul style="list-style-type: none"> ■ Discuss how timing and control signals are used in modern CPUs, such as in multi-core processors and advanced computing systems. ○ Case Study: <ul style="list-style-type: none"> ■ Walk through the execution of a simple instruction cycle, illustrating the role of timing and control signals at each stage. <p>5. YouTube Video:</p> <p>Exercise (5 minutes)</p> <ul style="list-style-type: none"> ● Ask students to summarize the different types of timing and control signals and their significance. ● Discuss the answers with the class.
<p>Closure</p>	<ul style="list-style-type: none"> ● Summarize the Lesson Learning Outcomes and get affirmation from students on these. ● Suggested Reading: <ul style="list-style-type: none"> ○ "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 120-145. ○ "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 3, pp. 85-112. <p>Homework:</p> <ul style="list-style-type: none"> ● Write a short essay on the role of control signals in modern computer processors and upload it on the learning management system.
<p>Evaluation</p>	<ul style="list-style-type: none"> ● How would you define timing and control signals in your own words? ● What are the basic types of control signals used in a CPU? ● Why are clock signals essential in digital systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 16	Course Name: Computer Organization and architecture Topic: Logic Micro Operations	Course No.:403
---------------------------	--	-----------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of logic micro operations. Identify different types of logic micro operations. Explain the significance of logic micro operations in computer systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>1. Introduction (5 minutes)</p> <ul style="list-style-type: none"> - What do you understand by the term "logic micro operations"? - How often do you encounter logical operations in computing tasks? - Explain the relevance of logic micro operations in the functioning of CPUs and ALUs. <p>2. Development (30 minutes)</p> <p>a. What are Logic Micro Operations?</p> <ul style="list-style-type: none"> -Definition: Explain logic micro operations as operations performed on the bits of a word, involving logical manipulation like AND, OR, XOR, and NOT. These are used in the arithmetic logic unit (ALU) of a computer. -Importance: Discuss how logic micro operations enable bitwise data manipulation and support various computational tasks. Highlight their role in decision-making processes within the CPU. <p>b. Types of Logic Micro Operations:</p> <ul style="list-style-type: none"> • AND Operation: <ul style="list-style-type: none"> -Outputs true if both corresponding bits of the operands are true. -Example: 1010 AND 1100 = 1000 • OR Operation: <ul style="list-style-type: none"> -Outputs true if at least one corresponding bit of the operands is true. -Example: 1010 OR 1100 = 1110 • XOR Operation: <ul style="list-style-type: none"> -Outputs true if only one of the corresponding bits is true. -Example: 1010 XOR 1100 = 0110 • NOT Operation: <ul style="list-style-type: none"> -Inverts the bits of the operand. -Example: NOT 1010 = 0101

	<p>c. Truth Tables:</p> <ul style="list-style-type: none"> -Provide truth tables for AND, OR, XOR, and NOT operations. -Discuss the output for all possible input combinations. <p>d. Applications of Logic Micro Operations:</p> <ul style="list-style-type: none"> ● Real-World Applications: <ul style="list-style-type: none"> -Discuss examples like error detection and correction, data masking, and bitwise manipulation in various algorithms. -Highlight the use of logic operations in encryption and compression algorithms. ● Case Study: <ul style="list-style-type: none"> -Walk through the design of a simple logic circuit using basic logic operations. <p>e. YouTube Video:</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to summarize the different types of logic micro operations and their functions. - Discuss the answers with the class.
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading <ul style="list-style-type: none"> - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 125-160. - "Computer System Architecture" by M. Morris Mano, Chapter 5, pp. 200-225.
Evaluation	<ol style="list-style-type: none"> 1. How would you define logic micro operations in your own words? 2. What are the different types of logic micro operations? 3. Why are logic micro operations essential in digital systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>



Lesson Plan No. 17	Course Name: Computer Networks Topic: Memory Reference Instructions	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of memory reference instructions. Identify different types of memory reference instructions and their functions. Explain the execution cycle of memory reference instructions. Recognize the significance of memory reference instructions in computer architecture.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> Start with a question: "What do you understand by the term 'memory reference instruction'?" Discuss the relevance of memory reference instructions in computer systems and their role in program execution. <p>Development (30 minutes)</p> <p>Memory Reference Instructions:</p> <ol style="list-style-type: none"> Definition: <ul style="list-style-type: none"> Explain that memory reference instructions are commands in machine language that direct the processor to perform operations involving memory locations. Types of Memory Reference Instructions: <ul style="list-style-type: none"> Load (LDA): Transfers data from a memory location to a register. Store (STA): Transfers data from a register to a memory location. Add (ADD): Adds data from a memory location to a register. Subtract (SUB): Subtracts data from a memory location to a register. Jump (JMP): Changes the sequence of instruction execution by setting the program counter to a specified address. Execution Cycle: <ul style="list-style-type: none"> Fetch: The instruction is fetched from memory. Decode: The control unit decodes the fetched instruction to determine the operation and the operands. Execute: The specified operation is performed, and the result is stored as needed. Importance in Computer Architecture: <ul style="list-style-type: none"> Discuss how memory reference instructions are crucial for data manipulation and control flow in programs.

	<ul style="list-style-type: none"> ○ Highlight their role in implementing algorithms and enabling complex computations. <p>Examples:</p> <ul style="list-style-type: none"> ● Provide example assembly code snippets to illustrate the use of memory reference instructions. ● Walk through the execution of each instruction step-by-step. <p>Exercise (5 minutes)</p> <ul style="list-style-type: none"> ● Ask students to write down the sequence of operations for a given memory reference instruction. ● Discuss the answers with the class and provide feedback.
Closure	<ul style="list-style-type: none"> ● Summarize the key points covered in the lesson. ● Ensure students understand the different types of memory reference instructions and their execution cycles. <p>Suggested Reading:</p> <ul style="list-style-type: none"> ● "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4. ● "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 5.
Evaluation	<ul style="list-style-type: none"> ● How would you define memory reference instructions in your own words? ● What are the different types of memory reference instructions, and what do they do? ● Describe the execution cycle of a memory reference instruction. <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 18	Course Name: Computer Networks Topic: Input-Output Interrupts	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of interrupts in computer systems. Identify different types of interrupts. Explain the working of input-output interrupts. Recognize the importance of interrupts in efficient computer operation.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ol style="list-style-type: none"> Ask students: What do you understand by the term "interrupt" in computer systems? Discuss real-life examples where interruptions are necessary and beneficial. Explain the relevance of interrupts in managing multiple tasks efficiently in computers. <p>Development (30 minutes)</p> <ol style="list-style-type: none"> What are Interrupts? <ul style="list-style-type: none"> Definition: Explain interrupts as signals that inform the CPU of an event that needs immediate attention, pausing the current CPU operations. Importance: Discuss how interrupts improve the efficiency of CPU operations by allowing multitasking and handling external events. Types of Interrupts: <ul style="list-style-type: none"> Hardware Interrupts: Triggered by hardware devices (e.g., keyboard, mouse). Software Interrupts: Generated by software instructions or programs. Maskable Interrupts: Can be turned off by the CPU to avoid interruption. Non-maskable Interrupts: Cannot be turned off and must be processed. Input-Output (I/O) Interrupts: <ul style="list-style-type: none"> Functioning: <ul style="list-style-type: none"> Describe how I/O interrupts allow peripheral devices to communicate with the CPU. Explain the steps: An I/O device signals an interrupt, the CPU pauses its current tasks, executes the interrupt service routine (ISR), and then resumes its previous tasks. Example: Explain the process with an example of a keyboard input interrupt.

	<p>4. Advantages of Using Interrupts:</p> <ul style="list-style-type: none"> ○ Efficient CPU utilization by allowing multitasking. ○ Faster response to external events compared to polling. ○ Reduced CPU idle time. <p>5. Case Study:</p> <ul style="list-style-type: none"> ○ Walk through a simple case where a printer uses interrupts to notify the CPU when it is ready to receive more data. <p>6. YouTube Video:</p> <p>Exercise (5 minutes)</p> <ul style="list-style-type: none"> ● Ask students to summarize the types of interrupts and their functions. ● Discuss their answers with the class to ensure understanding.
Closure	<p>Summarize the lesson's learning outcomes and confirm that students understand the key concepts of input-output interrupts.</p> <p>Suggested Reading:</p> <ul style="list-style-type: none"> ● "Computer Organization and Architecture" by William Stallings, Chapter 7, pp. 240-275. ● "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 4, pp. 150-190.
Evaluation	<ul style="list-style-type: none"> ● How would you define an interrupt in your own words? ● What are the basic types of interrupts? ● Why are input-output interrupts essential for computer operation? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 19	Course Name: Computer Networks Topic: Design of Basic Computer	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic design and components of a computer. Identify the main functional units of a computer and their roles. Explain the operation and interconnection of these units.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> Question to the class: What comes to your mind when you think of a computer? Discussion: How do you think a computer processes information? Relevance: Explain the importance of understanding computer design for anyone working with technology. <p>Development (30 minutes)</p> <ol style="list-style-type: none"> Overview of Basic Computer Design: <ul style="list-style-type: none"> Definition: <ul style="list-style-type: none"> A basic computer is an electronic device that processes data according to a set of instructions (program). Comprises various components working together to perform operations. Main Components of a Basic Computer: <ul style="list-style-type: none"> Central Processing Unit (CPU): <ul style="list-style-type: none"> Control Unit (CU): Directs operations of the computer. Arithmetic Logic Unit (ALU): Performs arithmetic and logical operations. Memory Unit: <ul style="list-style-type: none"> Primary Memory (RAM): Temporary storage for data and instructions. Secondary Memory: Permanent storage (e.g., hard drives, SSDs). Input and Output Devices: <ul style="list-style-type: none"> Input Devices: Devices like keyboard, mouse, etc., for data entry. Output Devices: Devices like monitors, printers, etc., for data output. Interconnection of Components: <ul style="list-style-type: none"> Buses: Communication pathways connecting CPU, memory, and I/O devices. <ul style="list-style-type: none"> Data Bus: Transfers data. Address Bus: Transfers information about where data should be sent or retrieved. Control Bus: Transfers control signals.

	<p>4. Instruction Set Architecture (ISA):</p> <ul style="list-style-type: none"> ○ Definition: The set of instructions a computer can execute. ○ Operation: How the CPU interprets and executes these instructions. ○ Examples: Machine language instructions for data movement, arithmetic operations, and control operations. <p>5. Basic Operations of a Computer:</p> <ul style="list-style-type: none"> ○ Fetch: Retrieving an instruction from memory. ○ Decode: Interpreting the instruction. ○ Execute: Performing the operation specified by the instruction. ○ Store: Writing the result back to memory if needed. <p>6. Case Study:</p> <ul style="list-style-type: none"> ○ Simple Computer Operation: <ul style="list-style-type: none"> ■ Example of a simple program execution cycle using the basic operations (fetch, decode, execute, store).
Closure	<ul style="list-style-type: none"> ● Summary: Recap the lesson learning outcomes. ● Confirmation: Get affirmation from students on their understanding of the key concepts. <p>Suggested Reading</p> <ul style="list-style-type: none"> ● "Computer Organization and Architecture" by William Stallings, Chapter 1, pp. 1-40. ● "Computer Systems: A Programmer's Perspective" by Randal E. Bryant and David R. O'Hallaron, Chapter 2, pp. 50-85.
Evaluation	<ul style="list-style-type: none"> ● How would you describe the role of the CPU in a computer? ● What are the key components of a computer's memory system? ● Why is the instruction set architecture important in computer design? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 20	Course Name: Computer Networks Topic: Design of Accumulator Logic	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of an accumulator in a computer system. Identify the role and function of accumulator logic in processing operations. Design a simple accumulator logic circuit. Apply accumulator logic in basic arithmetic operations.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<p>Introduction (5 minutes)</p> <ul style="list-style-type: none"> Begin with a brief discussion on what an accumulator is. <ul style="list-style-type: none"> "Who has heard the term 'accumulator' in the context of computer systems before?" "How do you think an accumulator functions in the execution of programs?" Explain the significance of accumulator logic in the CPU and its role in executing instructions efficiently. <p>Development (30 minutes)</p> <ol style="list-style-type: none"> What is an Accumulator? <ul style="list-style-type: none"> Definition: <ul style="list-style-type: none"> Describe an accumulator as a register in a CPU where intermediate arithmetic and logic results are stored. Highlight its role in reducing the time required to fetch data from memory. Importance: <ul style="list-style-type: none"> Discuss how an accumulator simplifies the architecture and increases the speed of the CPU by handling intermediate results directly. Explain its use in various arithmetic operations such as addition, subtraction, and logical operations. Design of Accumulator Logic: <ul style="list-style-type: none"> Basic Structure: <ul style="list-style-type: none"> Illustrate the basic structure of an accumulator with a diagram. Components include input lines, output lines, and control signals. Operation: <ul style="list-style-type: none"> Explain the operation of an accumulator with simple examples: <ul style="list-style-type: none"> Adding two numbers. Storing the result.

	<ul style="list-style-type: none"> <ul style="list-style-type: none"> <ul style="list-style-type: none"> ■ Retrieving and modifying the result for further operations. ○ Logic Circuit Design: <ul style="list-style-type: none"> ■ Step-by-step design of a simple accumulator circuit. ■ Show how logic gates (AND, OR, NOT) and flip-flops are used to construct the accumulator. ■ Explain the control logic needed to manage data flow into and out of the accumulator. 3. Applications of Accumulator Logic: <ul style="list-style-type: none"> ○ Real-World Examples: <ul style="list-style-type: none"> ■ Discuss the use of accumulators in simple processors and microcontrollers. ■ Highlight the role of accumulators in signal processing and embedded systems. ○ Case Study: <ul style="list-style-type: none"> ■ Walk through the design and simulation of an accumulator used in a basic CPU model. <p>YouTube Video:</p> <p>Exercise (5 minutes)</p> <ul style="list-style-type: none"> ● Ask students to design a simple accumulator circuit using logic gates and flip-flops. ● Discuss their designs and clarify any doubts.
Closure	<ul style="list-style-type: none"> ● Summarize the lesson's key points: the role of accumulators, their design, and their applications. ● Confirm students' understanding through a quick Q&A session. <p>Suggested Reading</p> <ul style="list-style-type: none"> ● "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 85-110. ● "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 4, pp. 130-150.
Evaluation	<ul style="list-style-type: none"> ● How would you define an accumulator in your own words? ● What are the primary components of accumulator logic? ● Why is accumulator logic essential in a CPU? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 21	Course Name: Computer Organization and Architecture Topic: Processor organization	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Understand the fundamental concepts of processor organization. b. Identify the components and their functions within a processor. c. Analyze the relationship between processor organization and computer architecture.
Teaching Aids (if any)	<ul style="list-style-type: none"> a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	<p>1) Introduction (5 minutes)</p> <ul style="list-style-type: none"> a. Ask questions to gauge prior knowledge: b. What is processor organization? c. Can you name any components of a processor? d. Discuss the importance of processor organization in computer architecture. <p>2) Development (30 minutes)</p> <ul style="list-style-type: none"> a. Explain the basic components of processor organization: b. Arithmetic Logic Unit (ALU), Control Unit, Registers, Memory Interface. c. Discuss the role of each component in instruction execution and data processing. <p>3) Exercise (5 minutes)</p> <ul style="list-style-type: none"> a. Binary to Decimal: Convert the binary number 101101101101 to its decimal equivalent. b. Decimal to Hexadecimal: Convert the decimal number 9292 to its hexadecimal representation. c. Octal to Binary: Convert the octal number 345345 to binary form.
Closure	<ul style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy. 3. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	<ul style="list-style-type: none"> 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on COA <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 22	Course Name: Computer Organization and Architecture Topic: General-purpose registers	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Understand the concept of general register organization. b. Identify the purpose and functions of general-purpose registers.
Teaching Aids (if any)	<ul style="list-style-type: none"> a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	<p>1) Introduction (5 minutes)</p> <ul style="list-style-type: none"> a. What are registers in a computer system? b. Why are registers important in computer organization? c. Define general register organization and its significance. d. Discuss the importance of general-purpose registers in processor operations. e. Provide examples of general-purpose registers in typical processor architectures. <p>2) Development (30 minutes)</p> <ul style="list-style-type: none"> a. Explain the concept of registers and their organization within a processor. b. Discuss the purpose of general-purpose registers, special-purpose registers, and program counter (PC). c. Introduce RTL as a symbolic language used to describe register transfer operations. d. Provide examples of RTL statements to demonstrate register operations. <p>3) Exercise (5 minutes)</p> <ul style="list-style-type: none"> a. Why are registers important in computer architecture? b. How do registers contribute to efficient instruction execution?
Closure	<ul style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy. 3. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	<ul style="list-style-type: none"> 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on COA <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 23	Course Name: Computer Organization and Architecture Topic: Stack organization	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Understand the concept of stack organization. b. Explain the role of a stack in computer architecture. c. Learn about stack operations and their implementations.
Teaching Aids (if any)	<ul style="list-style-type: none"> a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	<ol style="list-style-type: none"> 1) Introduction (5 minutes) <ul style="list-style-type: none"> a. Ask questions: <ul style="list-style-type: none"> b. What is stack organization? c. Why is stack organization important in computer architecture? d. How does a stack work in programming languages? e. Discuss the basic concept of stack organization and its significance. f. Introduce the concept of Last In First Out (LIFO) data structure. 2) Development (30 minutes) <ul style="list-style-type: none"> a. Basic Stack Operations: <ul style="list-style-type: none"> b. Push: Adding an item to the top of the stack. c. Pop: Removing an item from the top of the stack. d. Discuss how stacks are implemented in computer memory. 3) Exercise (5 minutes) <ul style="list-style-type: none"> a. Stack Operations: <ul style="list-style-type: none"> b. Perform push and pop operations on a stack with given elements.
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy. 3. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on COA <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 24	Course Name: Computer Organization and Architecture Topic: Addressing modes	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Define addressing modes in computer architecture b. Understand different types of addressing modes c. Analyze the use cases for various addressing modes.
Teaching Aids (if any)	<ul style="list-style-type: none"> d. Projector, Slides e. Use of Nearpod tool for online quiz
Teaching Development	<p>1) Introduction (5 minutes)</p> <ul style="list-style-type: none"> a. Start with questions to engage students: b. What is an addressing mode? c. Why are addressing modes important in computer architecture? d. Define addressing mode and its significance. <p>2) Development (30 minutes)</p> <ul style="list-style-type: none"> a. Basic Concepts of Addressing Modes: b. Direct Addressing Mode c. Indirect Addressing Mode d. Register Addressing Mode e. Immediate Addressing Mode f. Indexed Addressing Mode g. Relative Addressing Mode <p>3) Exercise (5 minutes)</p> <ul style="list-style-type: none"> a. Discuss scenarios for each addressing mode and ask students to identify the most suitable mode for each scenario.
Closure	<ul style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading 3. " Computer System Architecture," by Morris Mano. 4. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	<ul style="list-style-type: none"> 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on COA <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>

Lesson Plan No. 25	Course Name: Computer Organization and Architecture Topic: Booth's algorithm	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Understand the principles of binary multiplication. b. Comprehend Booth's algorithm and its significance in computer arithmetic. c. Implement Booth's algorithm for binary multiplication.
Teaching Aids (if any)	<ul style="list-style-type: none"> a. Projector, Slides b. Use of Nearpod tool for online quiz
Teaching Development	<ol style="list-style-type: none"> 1. Introduction (5 minutes) <ul style="list-style-type: none"> • Ask questions to gauge students' understanding of binary multiplication. • Define binary multiplication and its importance in computer arithmetic. • Briefly introduce Booth's algorithm and its historical context. 2. Development (30 minutes) <ol style="list-style-type: none"> a. Binary Multiplication Basics: <ul style="list-style-type: none"> • Explain the principles of binary multiplication. • Illustrate with simple examples. b. Mathematical Background: <ul style="list-style-type: none"> • Have a basic understanding of mathematical concepts particularly in arithmetic, as many topics involve mathematical operations. c. Booth's Algorithm: <ul style="list-style-type: none"> • Introduce Booth's algorithm, including its purpose and benefits. • Explain the algorithm step-by-step with a clear example. • Highlight the handling of positive and negative multipliers. 3. Exercise (5 minutes) <ul style="list-style-type: none"> • Provide a binary multiplication problem for students to solve using Booth's algorithm. • Encourage students to explain each step of the algorithm as they solve the problem.
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy (This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.) 3. "Digital Design" by M. Morris Mano and Michael D. Ciletti 4. Spend 5 minutes to wrap up and consolidate the learnings
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on COA

	Spend 5 minutes to evaluate student assimilation of the lesson contents
--	---

Lesson Plan No. 26	Course Name: Computer Organization and Architecture Topic: Division Algorithms	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ul style="list-style-type: none"> a. Understand the principles of division algorithms, including restoring and non-restoring methods. b. Comprehend the step-by-step process involved in both restoring and non-restoring division.
Teaching Aids (if any)	<ul style="list-style-type: none"> c. Projector, Slides d. Use of Nearpod tool for online quiz
Teaching Development	<ol style="list-style-type: none"> 1. Introduction (5 minutes) <ul style="list-style-type: none"> • Ask questions to assess prior knowledge about basic arithmetic operations. • Define division in the context of binary arithmetic. • Introduce the concept of division algorithms and differentiate between restoring and non-restoring division. 2. Development (30 minutes) <ul style="list-style-type: none"> • Understanding Division Algorithms: <ul style="list-style-type: none"> • Explain the basic concept of division in binary numbers. • Define and explain the steps involved in the restoring division algorithm. • Illustrate with a detailed example on how restoring division works. • Define and explain the steps involved in the non-restoring division algorithm. • Illustrate with a detailed example on how non-restoring division works. 3. Exercise (5 minutes) <ul style="list-style-type: none"> • Describe the process and significance of quotient and remainder in division algorithms. • Outline the steps in both restoring and non-restoring division algorithms with a simple binary example.
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy (This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.) 3. "Digital Design" by M. Morris Mano and Michael D. Ciletti 4. Additional online resources and video tutorials on division algorithms in computer architecture.
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on COA

	Spend 5 minutes to evaluate student assimilation of the lesson contents
--	---

Lesson Plan No. 27	Course Name: Computer Organization and Architecture Topic: Floating Point Representation	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> a. Understand the representation of floating-point numbers. b. Comprehend the arithmetic operations involving floating point numbers, including addition, subtraction, multiplication, and division.
Teaching Aids (if any)	<ol style="list-style-type: none"> c. Projector, Slides d. Use of Nearpod tool for online quiz
Teaching Development	<ol style="list-style-type: none"> 1. Introduction (5 minutes) <ul style="list-style-type: none"> • Ask questions to gauge the students' prior knowledge about floating point numbers. • Define floating point numbers and explain their significance in computer systems. • Introduce the IEEE 754 standard for floating point arithmetic. 2. Development (30 minutes) <ol style="list-style-type: none"> a. Understanding Floating Point Representation: <ul style="list-style-type: none"> • Explain the components of a floating-point number: sign, exponent, and mantissa (or significand). • Describe the normalization of floating-point numbers. b. Floating Point Arithmetic Operations: <ul style="list-style-type: none"> • Addition and Subtraction: • Aligning exponents, adding/subtracting significands, normalizing the result. • Multiplication and Division: <ul style="list-style-type: none"> • Multiplying/dividing significands, adding/subtracting exponents, normalizing the result. • Demonstrate examples of each operation step-by-step. 3. Exercise (5 minutes) <ul style="list-style-type: none"> • Provide a set of floating-point numbers and ask students to manually perform addition and multiplication operations.
Closure	<ol style="list-style-type: none"> 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these. 2. Suggested Reading "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy (This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.) 3. "Digital Design" by M. Morris Mano and Michael D. Ciletti 4. Spend 5 minutes to wrap up and consolidate the learnings.
Evaluation	<ol style="list-style-type: none"> 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss. 2. Nearpod Quiz on COA

	Spend 5 minutes to evaluate student assimilation of the lesson contents
--	---

Lesson Plan No. 27	Course Name: Computer Organization and Architecture Topic: Arithmetic operations	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the principles of decimal arithmetic operations in computer systems. Comprehend the implementation of decimal addition, subtraction, multiplication, and division in a CPU
Teaching Aids (if any)	<ol style="list-style-type: none"> Projector, Slides Use of Nearpod tool for online quiz
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Ask introductory questions about arithmetic operations. Define decimal arithmetic and its significance in computer systems. Development (30 minutes) <ul style="list-style-type: none"> Discuss the importance of decimal arithmetic in computing. Introduce the concepts of BCD (Binary-Coded Decimal) representation and its use in decimal arithmetic operations. Explain the process of performing decimal addition and subtraction. Illustrate with examples the step-by-step procedure of adding and subtracting decimal numbers using BCD. Exercise (5 minutes) <ul style="list-style-type: none"> Basic Concepts of Decimal Arithmetic: Describe how decimal arithmetic is handled in modern processors. Outline the steps for performing a BCD addition and identify potential challenges.
Closure	<ol style="list-style-type: none"> Summarize the Lesson Learning Outcomes and get affirmation from students on these. Suggested Reading "Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy (This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.) "Digital Design" by M. Morris Mano and Michael D. Ciletti Spend 5 minutes to wrap up and consolidate the learnings.
Evaluation	<ol style="list-style-type: none"> Reflective Questions (What, Why, Who?). Allow students to answer and discuss. Nearpod Quiz on COA <p>Spend 5 minutes to evaluate student assimilation of the lesson contents</p>



Lesson Plan No. 29	Course Name: Computer Organization and architecture Topic: Input/Output Interface	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of Input/Output (I/O) Interface. Identify and describe different types of I/O interfaces and their functions. Explain the role of I/O interfaces in computer systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Start by asking: "How do computers interact with external devices like keyboards, monitors, and printers?" Discuss the concept of input and output in computing and introduce the idea of an I/O interface as the bridge between the computer and external devices. Explain the importance of I/O interfaces in ensuring efficient communication between the computer and peripheral devices. Development (30 minutes) <ol style="list-style-type: none"> Overview of Input/Output Interface -Definition: Explain the I/O interface as the component that manages communication between the central processing unit (CPU) and peripheral devices. It serves as an intermediary that handles data transfer, commands, and status signals. Basic Components of an I/O Interface: I/O Ports: Describe I/O ports as hardware interfaces through which data is transferred between the CPU and peripherals. Explain the concept of memory-mapped I/O and isolated I/O. Data Registers: Explain how data registers within the I/O interface temporarily hold data being transferred between the CPU and peripherals. Types of I/O Interfaces <ul style="list-style-type: none"> Programmed I/O: Discuss how the CPU actively monitors the status of the I/O device and controls the data transfer process. Interrupt-driven I/O: Explain how the I/O device interrupts the CPU when it is ready to transfer data, allowing the CPU to perform other tasks in the meantime. Direct Memory Access (DMA): Describe DMA as a method where an I/O device can directly transfer data to or from

	<p>memory without involving the CPU, thereby speeding up the process.</p> <p>3. Exercise (5 minutes)</p> <ul style="list-style-type: none"> - Ask students to list the types of I/O interfaces discussed and briefly explain their functions. - Discuss their answers to ensure clarity and understanding..
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic. 2. Suggested Reading <ul style="list-style-type: none"> - "Computer System Architecture" by M. Morris Mano, Chapter 7, pp. 230-275. - "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 6, pp. 260-300.
Evaluation	<ol style="list-style-type: none"> 1. How would you define an I/O interface in your own words? 2. What are the different types of I/O interfaces, and how do they differ in operation? 3. Why are I/O interfaces essential in computer systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 30	Course Name: Computer Organization and architecture Topic: Asynchronous Data Transfer	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of asynchronous data transfer. Identify and describe the methods and protocols used in asynchronous data transfer. Explain the significance of asynchronous data transfer in computer systems.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Begin by asking: "What do you understand by the term 'Asynchronous Data Transfer'?" Discuss how asynchronous data transfer differs from synchronous data transfer. Explain the relevance of asynchronous data transfer in handling communication between devices that operate at different speeds. Development (30 minutes) <ol style="list-style-type: none"> Overview of asynchronous data transfer -Definition: Define asynchronous data transfer as a method of data communication where data is sent without a fixed timing mechanism, with each byte or group of bytes typically preceded by a start bit and followed by a stop bit. Importance of Asynchronous Data Transfer: Discuss how asynchronous data transfer allows devices with different clock rates to communicate effectively. Highlight its role in serial communication protocols such as UART (Universal Asynchronous Receiver/Transmitter). Applications of Asynchronous Data Transfer: <ul style="list-style-type: none"> Real-World Applications: Discuss examples such as RS-232 serial communication and USB (Universal Serial Bus) interfaces. Highlight the use of asynchronous transfer in modem communications and other telecommunication systems. Exercise (5 minutes) <ul style="list-style-type: none"> Ask students to summarize the key components of asynchronous data transfer and their functions.

	- Discuss the answers with the class to ensure understanding.
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic. 2. Suggested Reading <ul style="list-style-type: none"> - "Computer Networks" by Andrew S. Tanenbaum, Chapter 2, pp. 45-90. - "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 6, pp. 275-320.
Evaluation	<ol style="list-style-type: none"> 1. How would you define asynchronous data transfer in your own words? 2. What are the key components of asynchronous data transfer? 3. Why is asynchronous data transfer essential in digital communication systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 31	Course Name: Computer Organization and architecture Topic: Mode of Data Transfer	Course No.: COM-301
---------------------------	---	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concepts of different modes of data transfer. Identify and describe various methods used for transferring data between devices. Explain the significance of each mode in terms of speed, efficiency, and application.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Begin by asking: "How do computers transfer data between devices, such as memory and I/O devices?" Discuss the importance of data transfer in computer systems and how it affects the performance of the system. Explain how different data transfer modes are suited for different tasks. Development (30 minutes) <ol style="list-style-type: none"> Overview of Data Transfer -Definition: Define data transfer as the movement of data between two or more components in a computer system. Basic Components of Data Transfer: <ul style="list-style-type: none"> Programmed I/O: Discuss how the CPU actively monitors the status of the I/O device and controls the data transfer process. Interrupt-driven I/O: Explain how the I/O device interrupts the CPU when it is ready to transfer data, allowing the CPU to perform other tasks in the meantime. Direct Memory Access (DMA): Describe DMA as a method where an I/O device can directly transfer data to or from memory without involving the CPU, thereby speeding up the process. Exercise (5 minutes) <ul style="list-style-type: none"> Ask students to summarize the key differences between the various modes of data transfer. Discuss the answers with the class to ensure understanding.
Closure	<ol style="list-style-type: none"> Summarize the lesson learning outcomes and ask students for their feedback on the topic. Suggested Reading

	<ul style="list-style-type: none"> - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 7, pp. 320-350. - "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 4, pp. 250-290.
Evaluation	<ol style="list-style-type: none"> 1. How would you describe the different modes of data transfer in your own words? 2. What are the advantages of using DMA over programmed I/O? 3. Why is interrupt-driven I/O preferred in real-time systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 32	Course Name: Computer Organization and architecture Topic: Priority Interrupt	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the concept of priority interrupt and its significance in computer systems. Identify different types of interrupts and their priority levels. Explain how priority interrupt mechanisms are implemented in hardware.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> "What happens when multiple devices need the CPU's attention simultaneously?" Discuss the concept of interrupts and why prioritizing them is essential for system efficiency. Introduce the term "priority interrupt" and its relevance in computer architecture. Development (30 minutes) <ol style="list-style-type: none"> What is Priority Interrupt? -Definition: Explain that a priority interrupt is a system that manages multiple interrupts by assigning them different priority levels. Highlight that the CPU responds to the highest priority interrupt first, ensuring critical tasks are handled promptly. Types of Interrupts: <ul style="list-style-type: none"> External Interrupts Internal Interrupts Software Interrupts Importance of Priority Interrupt: <ul style="list-style-type: none"> -Discuss the role of priority interrupt in ensuring system reliability and efficiency. -Highlight how it prevents low-priority tasks from blocking high-priority ones, improving overall system performance.. Exercise (5 minutes) <ul style="list-style-type: none"> Ask students to summarize the key components of asynchronous data transfer and their functions. Discuss the answers with the class to ensure understanding.



Kot Bhalwal, Jammu



Closure	<ol style="list-style-type: none">1. Summarize the lesson learning outcomes and ask students for their feedback on the topic.2. Suggested Reading<ul style="list-style-type: none">- "Computer Networks" by Andrew S. Tanenbaum, Chapter 2, pp. 45-90.- "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 6, pp. 275-320.
Evaluation	<ol style="list-style-type: none">1. How would you define asynchronous data transfer in your own words?2. What are the key components of asynchronous data transfer?3. Why is asynchronous data transfer essential in digital communication systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 33	Course Name: Computer Organization and architecture Topic: Direct Memory Access (DMA)	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concept of Direct Memory Access (DMA). Identify and describe the role of DMA in data transfer within computer systems. Explain how DMA improves system performance by bypassing the CPU during data transfer.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> What are the traditional methods of data transfer between memory and peripherals?" Discuss the limitations of CPU-driven data transfer (e.g., Programmed I/O and Interrupt-driven I/O). Introduce DMA as a method that allows peripherals to communicate directly with memory, bypassing the CPU. Development (30 minutes) <ol style="list-style-type: none"> What is Direct Memory Access? -Definition: Explain DMA as a feature that allows peripherals to directly read from and write to the main memory, reducing the CPU's workload. Basic Components of DMA: <ul style="list-style-type: none"> DMA Controller DMA Channels Software Interrupts Importance of Priority Interrupt: -Performance Improvement: Discuss how DMA increases system efficiency by allowing data transfer without burdening the CPU. Highlight the reduced latency in data transfer, especially in high-speed applications. Exercise (5 minutes) <ul style="list-style-type: none"> Ask students to summarize the advantages of using DMA in system design. Discuss the answers with the class to ensure understanding.
Closure	1. Summarize the lesson learning outcomes and ask students for their feedback on the topic.

	<p>2. Suggested Reading</p> <ul style="list-style-type: none"> - "Computer Organization and Design" by David Patterson and John Hennessy, Chapter 6, pp. 300-350. - "Modern Computer Architecture" by John Paul Shen, Chapter 7, pp. 150-200.
Evaluation	<ol style="list-style-type: none"> 1. How does DMA improve system performance compared to traditional data transfer methods? 2. What are the key components of a DMA system? 3. Explain a real-world application where DMA is crucial. <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 34	Course Name: Computer Organization and architecture Topic: Input-Output Processor (IOP)	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concept of an Input-Output Processor (IOP). Explain the need for an IOP in a computer system. Describe the functioning and structure of an IOP. Recognize the difference between an IOP and a central processing unit (CPU).
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Begin by asking: "What do you understand by the term 'Input-Output Processor'?" Discuss the role of input-output operations in computer systems. Explain the necessity of offloading input-output tasks from the CPU to an IOP for better efficiency and performance. Development (30 minutes) <ol style="list-style-type: none"> Definition of Input-Output Processor (IOP) -Definition: Explain that the IOP is a specialized processor designed to handle input-output tasks, reducing the load on the CPU. Basic Components of an Input-Output Processor: <ul style="list-style-type: none"> Control Unit Registers Interface Logic Functions of the IOP: Explain how the IOP performs I/O tasks independently of the CPU, handling direct memory access (DMA), interrupts, and data buffering. Provide examples such as transferring data between the hard drive and main memory without CPU intervention. Exercise (5 minutes). <ul style="list-style-type: none"> Ask students to summarize the role of an Input-Output Processor and its components. Engage in a brief discussion to ensure the students have understood the concepts.
Closure	<ol style="list-style-type: none"> Summarize the lesson learning outcomes and ask students for their feedback on the topic.

	<p>2. Suggested Reading</p> <ul style="list-style-type: none"> - "Computer Organization and Design" by David Patterson and John Hennessy, Chapter 6, pp. 300-350. - "Modern Computer Architecture" by John Paul Shen, Chapter 7, pp. 150-200.
Evaluation	<ol style="list-style-type: none"> 1. How would you define an Input-Output Processor in your own words? 2. What are the basic functions of an IOP? 3. Why is an IOP beneficial in high-performance systems? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 35	Course Name: Computer Organization and architecture Topic: Memory Hierarchy	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept and significance of memory hierarchy in a computer system. Identify and describe the different levels of memory hierarchy (registers, cache, main memory, secondary storage). Explain the characteristics and performance trade-offs at each level of the memory hierarchy.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> "Why do we have different types of memory in a computer system?" Introduce the concept of memory hierarchy, highlighting the need for different levels of memory to balance speed and cost. Explain that memory hierarchy is a structured way to manage data storage and access efficiently within a computer. Development (30 minutes) <ol style="list-style-type: none"> Definition of Memory Hierarchy -Definition:Memory hierarchy is a layered structure that categorizes memory into levels based on speed, cost, and size. Levels of Memory Hierarchy <ul style="list-style-type: none"> Registers Cache Memory Main Memory (RAM) Secondary Storage (Hard Drives, SSDs) Characteristics and Performance at Each Level Compare the characteristics of memory at each level in terms of access time, size, cost, and purpose. Discuss how the memory hierarchy improves the overall performance of a system by ensuring that the CPU can access data at different speeds and sizes. Exercise (5 minutes). <ul style="list-style-type: none"> Ask students to summarize the different levels of memory hierarchy and their key characteristics. Discuss the answers with the class to ensure understanding.

Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic. 2. Suggested Reading <ul style="list-style-type: none"> - "Computer Organization and Design" by David Patterson and John Hennessy. - "Modern Computer Architecture" by John Paul Shen.
Evaluation	<ol style="list-style-type: none"> 1. How would you explain the concept of memory hierarchy? 2. What are the key differences between cache memory and main memory? 3. Why is the memory hierarchy essential for CPU performance? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 36	Course Name: Computer Organization and architecture Topic: Cache Memory	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: <ol style="list-style-type: none"> Understand the basic concept of cache memory and its significance in computer architecture. Explain the different types of cache memory and their functionality. Recognize the importance of cache mapping techniques and cache performance measures
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> What is cache memory, and why do we need it in computer systems?" Briefly explain how cache memory is used to reduce the time to access data from the main memory. Highlight its importance in speeding up the performance of a computer system. Development (30 minutes) <ol style="list-style-type: none"> Definition of Cache Memory -Definition: Explain cache memory as a smaller, faster type of volatile computer memory that provides high-speed data access to the processor. Types of Cache Memory <ul style="list-style-type: none"> L1 Cache: Located within the processor itself; it is the smallest and fastest type of cache. L2 Cache: Larger than L1 and may be located on the CPU chip or close to it. L3 Cache: Larger and slower than L1 and L2 but still faster than main memory, shared across cores in multi-core processors. Cache Mapping Techniques: <p>Direct Mapping: Each block of main memory maps to exactly one cache line.</p> <p>Associative Mapping: Any block of main memory can map to any line of cache.</p>

	<p>Set-Associative Mapping: Combines elements of both direct and associative mapping. Blocks are mapped to a set, and then within that set, blocks are placed anywhere.</p> <p>3. Exercise (5 minutes).</p> <ul style="list-style-type: none"> - Ask students to summarize the different types of cache and their functions. - Discuss their responses and clarify any misconceptions.
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic. 2. Suggested Reading <ul style="list-style-type: none"> - "Computer Organization and Design" by David Patterson and John Hennessy. - "Modern Computer Architecture" by John Paul Shen.
Evaluation	<ol style="list-style-type: none"> 1. Define cache memory in your own words. 2. What are the different types of cache memory? 3. How do cache mapping techniques affect cache performance? 4. How can cache memory improve the overall system performance? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>

Lesson Plan No. 37	Course Name: Computer Organization and architecture Topic: Virtual Memory	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	<p>At the end of the lesson the student shall be able to:</p> <ol style="list-style-type: none"> Understand the concept of virtual memory and its significance in computer systems. Explain how virtual memory manages memory hierarchy and improves performance. Describe the process of paging and segmentation in virtual memory. Recognize the role of the Memory Management Unit (MMU) in translating virtual addresses to physical addresses.
Teaching Aids (if any)	<ol style="list-style-type: none"> Presentation slides. Projector You Tube video
Teaching Development	<ol style="list-style-type: none"> Introduction (5 minutes) <ul style="list-style-type: none"> Begin by asking: "Have you ever wondered how computers run programs larger than their physical memory?" Explain the concept of memory hierarchy and introduce virtual memory as a technique to extend physical memory using disk space. Discuss the relevance of virtual memory in modern operating systems. Development (30 minutes) <ol style="list-style-type: none"> Definition of Virtual Memory -Definition: Virtual memory is a memory management technique that allows programs to use more memory than is physically available by using disk space as an extension of RAM. Key Components of Virtual Memory: <ul style="list-style-type: none"> ● Paging ● Segmentation ● Memory Management Unit (MMU) ● Page Table Benefits of Virtual Memory Increased Multitasking: More programs can run simultaneously.

	<p>Security and Isolation: Each process has its own virtual memory, preventing interference.</p> <p>3. Exercise (5 minutes).</p> <ul style="list-style-type: none"> - Ask students to summarize the different types of cache and their functions. - Discuss their responses and clarify any misconceptions.
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic. 2. Suggested Reading <ul style="list-style-type: none"> - "Computer Organization and Design" by David Patterson and John Hennessy. - "Modern Computer Architecture" by John Paul Shen.
Evaluation	<ol style="list-style-type: none"> 1. How would you define virtual memory in your own words? 2. What is the purpose of the Memory Management Unit (MMU)? 3. Can you explain the process of handling a page fault? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>



Kot Bhalwal, Jammu



Lesson Plan No. 38	Course Name: Computer Organization and architecture Topic: Memory Management System	Course No.: COM-301
---------------------------	--	----------------------------

Objectives	At the end of the lesson the student shall be able to: a. Understand the fundamental concepts of memory management in operating systems. b. Identify and describe different memory allocation techniques such as paging, segmentation, and virtual memory. c. Explain the role of memory management in resource allocation and process management.
Teaching Aids (if any)	a. Presentation slides. b. Projector c. You Tube video
Teaching Development	1. Introduction (5 minutes) – What do you think happens to the memory when you run multiple programs on your computer?" – Discuss the importance of memory management in ensuring that the operating system efficiently allocates and deallocates memory to different processes. – Introduce the concept of memory management as a critical component of the operating system that oversees the usage of RAM and virtual memory. 2. Development (30 minutes) a. Definition of Memory Management -Define memory management as the process of coordinating and controlling computer memory, assigning portions to various running programs to optimize overall system performance. b. Memory Allocation Techniques <ul style="list-style-type: none">● Contiguous Memory Allocation● Segmentation● Paging● Paging c. Role of Memory Management in Resource Allocation -Discuss how memory management ensures that different processes get the memory they need while avoiding conflicts through techniques like swapping and dynamic loading. -Highlight the concept of memory protection, ensuring that one process does not affect the memory allocated to another process. 3. Exercise (5 minutes).

	<ul style="list-style-type: none"> - Ask students to identify and explain the difference between paging and segmentation. - Discuss the answers with the class to ensure comprehension of the memory management techniques.
Closure	<ol style="list-style-type: none"> 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic. 2. Suggested Reading <ul style="list-style-type: none"> - "Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, Chapter 8, pp. 300-350. - "Modern Operating Systems" by Andrew S. Tanenbaum, Chapter 3, pp. 120-170.
Evaluation	<ol style="list-style-type: none"> 1. How would you describe memory management in operating systems? 2. What is the difference between paging and segmentation? 3. Why is virtual memory important for running large applications? <p>Spend 5 minutes to wrap up and consolidate the learnings.</p>