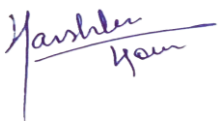
## Department of CSE

## Details of Lesson Plan

| S.No. | Particulars | Details |
|---|---|---|
| 1. | Course Name | Computer Organization and Architecture |
| 2. | Course Code | COM-403 |
| 3. | Academic Year | 2023-24 |
| 4. | Semester | 4th |
| 5. | Number of Lesson plans | 38 |
| 6. | Faculty Assigned | Ms. Harashleen Kour |



Faculty Signature

| Lesson Plan No. 1 | Course Name: Computer Organization and Architecture<br>Topic: Overview of Digital Electronics | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of digital electronics.<br>b. Identify different types of digital logic gates and their functions.<br>c. Explain the significance of binary numbers in digital electronics.<br>d. Recognize the applications of digital fundamentals in modern technology. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– What do you understand by the term "digital electronics"?<br>– How often do you encounter digital devices in your daily life?<br>– Explain the relevance of digital fundamentals in today's technology-driven world.<br><br>**2. Development (30 minutes)**<br>**a. What are Digital Fundamentals?**<br>**-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.<br>**-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.<br><br>**b. Basic Components of Digital Fundamentals:**<br>-Binary Numbers: The foundation of digital systems, where data is represented using two symbols (0 and 1).<br>-Logic Gates: The building blocks of digital circuits that perform logical operations.<br>-AND Gate: Outputs true only if all inputs are true.<br>-OR Gate: Outputs true if at least one input is true.<br>-NOT Gate: Inverts the input signal.<br>-NAND, NOR, XOR, XNOR Gates: Variations of basic logic gates with specific functions.<br>-Truth Tables: Tables that define the output of a logic gate for all possible input combinations.<br>-Boolean Algebra: Mathematical notation used to describe logic gate functions and simplify digital circuits. |

| | |
|---|---|
| | **c. Applications of Digital Fundamentals:**<br>• **Real-World Applications**<br>-Discuss examples like digital clocks, calculators, and computer processors.<br>-Highlight the use of digital logic in modern devices such as smartphones and smart home systems.<br><br>**d. Case Study:**<br>-Walk through the design of a simple digital circuit using basic logic gates.<br><br>**e. YouTube Video:**<br><br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the basic components of digital fundamentals and their functions.<br>- Discuss the answers with the class. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>- "Digital Design" by M. Morris Mano, Chapter 1, pp. 1-35.<br>- "Fundamentals of Digital Logic with Verilog Design" by Stephen Brown and Zvonko Vranesic, Chapter 2, pp. 40-65.<br>3. Homework<br>- Write a short essay on the importance of binary numbers in digital electronics and upload it on the learning Camu. |
| **Evaluation** | 1. How would you define digital fundamentals in your own words?<br>2. What are the basic components of digital electronics?<br>3. Why are binary numbers essential in digital systems?<br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 2 | Course Name: Computer Organization and Architecture<br>Topic: Basics of Computer Organization and Architecture | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of computer organization and architecture.<br>b. Identify the key components of a computer system and their functions.<br>c. Explain the significance of different computer architectures.<br>d. Recognize the applications of computer organization and architecture in modern computing. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– What do you understand by the term "computer organization and architecture"?<br>– How often do you think about the internal components when using a computer?<br>– Explain the relevance of understanding computer architecture in today's technology-driven world.<br><br>Development **(30 minutes)**<br>**a. Computer Organization and Architecture: Definition**<br>-Explain computer organization as the operational units and their interconnections that realize the architectural specifications.<br>-Explain computer architecture as the attributes of a system visible to a programmer, such as instruction sets, data formats, and addressing modes.<br>**b. Importance of Computer Organization and Architecture**<br>-Discuss how these concepts enable the design and functionality of computers.<br>-Highlight their role in the performance, efficiency, and capabilities of computer systems.<br>**c. Basic Components of Computer Systems:**<br>-Central Processing Unit (CPU): The brain of the computer that executes instructions.<br>-Memory:<br>-Primary Memory (RAM): Temporary storage for currently used data and instructions.<br>-Secondary Memory: Long-term storage for data and programs. |

| | |
|---|---|
| | -Input/Output Devices (I/O): Interfaces for the computer to interact with the external environment (e.g., keyboard, mouse, monitor).<br>-System Bus: The communication pathway connecting various components of the computer.<br>**d. Computer Architectures:**<br>• **Von Neumann Architecture:**<br>-Discuss its characteristics such as shared program and data memory.<br>• **Harvard Architecture:**<br>-Explain its separate storage and signal pathways for -instructions and data.<br>**e. Real-World Applications:**<br>-Discuss examples such as personal computers, smartphones, and servers.<br>-Highlight the use of different architectures in various devices.<br>**f. Case Study:**<br>-Walk through the design of a simple computer system using the Von Neumann architecture.<br>**g. YouTube Video:**<br>**2. Exercise (5 minutes)**<br>– Ask students to summarize the basic components of computer systems and their functions.<br>– Discuss the answers with the class. |
| **Closure** | **1. Closure**<br>– Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>**2. Suggested Reading**<br>– "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 1, pp. 1-35.<br>– "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 2, pp. 40-65 |
| **Evaluation** | 1. How would you define computer organization and architecture in your own words?<br>2. What are the basic components of a computer system?<br>3. Why is it important to understand different computer architectures?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 3 | Course Name: Computer Organization and Architecture<br>Topic: Conversions | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the fundamental concepts of number system conversions.<br>b. Convert numbers between different bases (binary, decimal, octal, and hexadecimal).<br>c. Explain the significance of various number systems in computer architecture.<br>d. Apply conversion techniques in practical computing scenarios. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– Ask students: "What number systems are you familiar with?"<br>– Discuss the importance of different number systems in computing.<br>– Explain the relevance of understanding conversions in computer architecture.<br>**2. Development (30 minutes)**<br>**a. Number Systems Overview**<br>-Definition: Explain the concept of number systems and their bases (e.g., binary is base-2, decimal is base-10).<br>-Importance: Discuss how computers use different number systems for various operations.<br>**b. Binary System (Base-2)**<br>-Conversion to Decimal: Explain the method of converting binary numbers to decimal.<br>-Example: Convert 110121101_211012 to decimal.<br>-Conversion from Decimal: Explain the method of converting decimal numbers to binary. Example: Convert 131013_{10}1310 to binary.<br>**c. Octal System (Base-8)**<br>-Conversion to Decimal: Explain the method of converting octal numbers to decimal.<br>-Example: Convert 1578157_81578 to decimal.<br>-Conversion from Decimal: Explain the method of converting decimal numbers to octal.<br>-Example: Convert 11110111_{10}11110 to octal.<br>**d. Hexadecimal System (Base-16)**<br>-Conversion to Decimal: Explain the method of converting hexadecimal numbers to decimal.<br>-Example: Convert 1A3161A3_{16}1A316 to decimal.<br>-Conversion from Decimal: Explain the method of converting decimal numbers to hexadecimal. |

| | |
|---|---|
| | -Example: Convert 41910419_{10}41910 to hexadecimal.<br>**e. Inter-conversion between Non-decimal Bases**<br>-Explain methods for converting directly between binary, octal, and hexadecimal.<br>-Example: Convert 1101012110101_21101012 to octal and hexadecimal.<br>**f. Applications in Computing**<br>-Discuss the role of different number systems in computer memory addressing, instruction sets, and data representation.<br><br>**3. Exercise (5 minutes)**<br>● **Ask students to perform a few conversion exercises:**<br>-Convert 101021010_210102 to decimal.<br>-Convert 25510255_{10}25510 to hexadecimal.<br>-Convert 3778377_83778 to binary. |
| **Closure** | 1. Summarize the key points of number system conversions.<br>2. Ensure students can explain the conversion processes and their importance in computing.<br>3. Suggested Reading<br>- "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 60-85.<br>- "Digital Logic and Computer Design" by M. Morris Mano, Chapter 1, pp. 10-30. |
| **Evaluation** | 1. How would you define number system conversions?<br>2. Convert 101021010_210102 to decimal.<br>3. Why is it important to understand different number systems in computer architecture?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 4 | Course Name: Computer Organization and Architecture<br>Topic: Logic Gates | Course No.: COM- |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of logic gates.<br>b. Identify different types of logic gates and their functions.<br>c. Explain the significance of logic gates in digital electronics.<br>d. Recognize the applications of logic gates in modern technology. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>- What do you understand by the term "logic gates"?<br>- How often do you encounter digital devices that use logic gates in your daily life?<br>- Introduce logic gates as the fundamental building blocks of digital circuits.<br>- Explain the relevance of logic gates in today's technology-driven world.<br>**2. Development (30 minutes)**<br>**a. What are Logic Gates?**<br>-Definition: Define logic gates as electronic devices that perform logical operations on one or more binary inputs to produce a single binary output.<br>-Importance: Discuss how logic gates enable the operation and functionality of various digital circuits and systems.<br>**b. Types of Logic Gates:**<br>**-AND Gate:** Outputs true (1) only if all inputs are true (1).<br>-Truth Table: |

| A | B | Output |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**-OR Gate:** Outputs true (1) if at least one input is true (1).
-Truth Table:

| 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:

| | |
|---|---|
| | <table><tr><th>A</th><th>B</th><th>Output</th></tr><tr><td>0</td><td>0</td><td>1</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></table><br>**c. Boolean Algebra:**<br>-Explanation: Introduce Boolean algebra as a mathematical notation used to describe logic gate functions and simplify digital circuits.<br>-Examples: Demonstrate simple Boolean expressions and their corresponding logic gate implementations.<br>**d. Applications of Logic Gates:**<br>-Real-World Applications: Discuss examples like digital clocks, calculators, and computer processors.<br>-Highlight the use of logic gates in modern devices such as smartphones and smart home systems.<br>**e. Case Study:**<br>-Walk through the design of a simple digital circuit using basic logic gates.<br>**f. YouTube Video:**<br>**3. Exercise (5 minutes)**<br>● **Activity:**<br>- Ask students to summarize the basic components of logic gates and their functions.<br>- Discuss the answers with the class. |
| **Closure** | 1. Recap the key points covered in the lesson.<br>2. Get affirmation from students on their understanding of the lesson.<br>3. Suggested Reading<br>- "Digital Design" by M. Morris Mano, Chapter 1, pp. 1-35.<br>- "Fundamentals of Digital Logic with Verilog Design" by Stephen Brown and Zvonko Vranesic, Chapter 2, pp. 40-65. |
| **Evaluation** | 1. How would you define logic gates in your own words?<br>2. What are the basic types of logic gates?<br>3. Why are logic gates essential in digital systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 5 | Course Name: Computer Organization and Architecture<br>Topic: Logic Gates | Course No.: COM- |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of complements in COA.<br>b. Differentiate between various types of complements.<br>c. Apply the concept of complements in arithmetic operations. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>**Discussion Questions:**<br>-What do you understand by the term "complement" in mathematics?<br>-Can you recall any scenarios where you have used complements?<br>-Explain the importance of complements in computer arithmetic, particularly in simplifying the subtraction operation and representing negative numbers.<br><br>**2. Development (30 minutes)**<br>• **Definition:**<br>-Explain complements as methods to transform numbers to facilitate arithmetic operations in digital systems.<br>• **Types of Complements:**<br>**1's Complement:**<br>-Definition: The 1's complement of a binary number is obtained by inverting all bits (changing 0s to 1s and 1s to 0s).<br>-Example: For binary number 1010, the 1's complement is 0101.<br>**2's Complement:**<br>-Definition: The 2's complement of a binary number is obtained by adding 1 to the 1's complement of the number.<br>-Example: For binary number 1010, the 1's complement is 0101, and the 2's complement is 0110.<br>• **Applications of Complements:**<br>**-Arithmetic Operations:** Simplifying subtraction by converting it to addition using 2's complement.<br>-Representing negative numbers in binary systems.<br>**-Problem Solving:** Example problems demonstrating subtraction using 2's complement.<br>-Illustrate how complements make it easier to perform binary arithmetic operations in digital systems.<br>• **YouTube Video:**<br>**3. Exercise (5 minutes)** |

| | |
|---|---|
| | - Ask students to compute the 1's and 2's complement of a given binary number.<br>- Discuss the solutions and ensure understanding. |
| **Closure** | 1. Recap the concepts of 1's and 2's complements, their calculation, and applications.<br>2. Suggested Reading<br>- "Computer System Architecture" by M. Morris Mano, Chapter 3, pp. 50-75.<br>- "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 2, pp. 30-60. |
| **Evaluation** | 1. How would you define 1's and 2's complement in your own words?<br>2. What are the steps to calculate the 2's complement of a binary number?<br>3. Why are complements essential in computer arithmetic?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 1 | Course Name: Computer Organization and architecture<br>Topic: Register Transfer Language | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of Register Transfer Language (RTL).<br>b. Identify and describe the different types of registers used in RTL.<br>c. Explain the significance of micro-operations and how they are represented in RTL.<br>d. Recognize the applications of RTL in the design and functioning of digital systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>- Begin by asking: "What do you understand by the term 'Register Transfer Language'?"<br>- Discuss how RTL is encountered in computer architecture and its importance in digital system design.<br>- Explain the relevance of RTL in understanding the operations within a computer processor.<br><br>**2. Development (30 minutes)**<br>**a. What is Register Transfer Language?**<br>● **Definition:**<br>-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.<br><br>**b. Basic Components of Register Transfer Language:**<br>● **Registers:**<br>-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.<br>● **Micro-operations:**<br>-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.<br>-Provide symbolic representations for common micro-operations, such as: |

| | |
|---|---|
| | -Data transfer: R2 <- R1 (Transfer data from register R1 to register R2)<br>-Addition: R3 <- R1 + R2 (Add the contents of registers R1 and R2 and store the result in R3)<br><br>**c. Importance of Register Transfer Language:**<br>-Discuss how RTL helps in designing and understanding the internal functioning of digital systems, especially in the development of microprocessors and control units.<br>-Highlight the role of RTL in simplifying the design process by providing a clear and concise way to represent complex operations.<br><br>**d. Applications of Register Transfer Language:**<br>● **Real-World Applications:**<br>-Discuss examples like the design of arithmetic logic units (ALUs), control units, and data paths in microprocessors.<br>-Highlight the use of RTL in hardware description languages (HDLs) such as VHDL and Verilog for designing and simulating digital circuits.<br>● **Case Study:**<br>-Walk through the design of a simple arithmetic unit using RTL notation. Explain each step and the corresponding RTL representation.<br>● **YouTube Video:**<br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the basic components of RTL and their functions.<br>- Discuss the answers with the class to ensure understanding. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading:<br>- "Computer System Architecture" by M. Morris Mano, Chapter 4, pp. 130-175.<br>- "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 5, pp. 200-250. |
| **Evaluation** | 1. How would you define Register Transfer Language in your own words?<br>2. What are the basic components of RTL?<br>3. Why are micro-operations essential in digital systems?<br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 2 | Course Name: Computer Organization and architecture<br>Topic: Register transfer | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of register transfer in computer architecture.<br>b. Identify different types of registers and their functions.<br>c. Explain the significance of register transfer in the execution of instructions.<br>d. Recognize the applications of register transfer in computer systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>- What do you understand by the term "register transfer"?<br>- How often do you think registers are used in a computer's operation?<br>- Explain the relevance of register transfer in the functioning of computer systems.<br><br>**2. Development (30 minutes)**<br>**a. What is Register Transfer?**<br>● **Definition:**<br>-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.<br>● **Importance:**<br>-Discuss how register transfers are fundamental to the CPU's operation, enabling data processing, arithmetic operations, and instruction execution.<br><br>**b. Basic Components of Register Transfer:**<br>-Registers: Define registers and explain their purpose as temporary storage locations for data and instructions.<br>-Register Transfer Language (RTL): Explain RTL as a symbolic notation used to describe the operations, data flow, and control signals involved in register transfers.<br>-Micro-operations: Describe micro-operations as the fundamental operations performed on data stored in registers, such as data transfer, arithmetic operations, and logical operations. |

| | |
|---|---|
| | -Control Signals: Discuss the role of control signals in coordinating the movement of data between registers and the execution of micro-operations.<br>**c. Types of Registers:**<br>-General-Purpose Registers: Explain their use in storing temporary data and intermediate results during instruction execution.<br>-Special-Purpose Registers: Introduce examples like the Program Counter (PC), Instruction Register (IR), and Memory Address Register (MAR), explaining their specific functions.<br>**d. Applications of Register Transfer:**<br>-Real-World Applications: Discuss examples such as CPU operations, data processing tasks, and the functioning of modern computing devices.<br>-Case Study: Walk through a simple example of register transfer in a basic CPU instruction cycle, explaining each step and the registers involved.<br>**e. YouTube Video:**<br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the basic components and types of registers involved in register transfer.<br>- Discuss the answers with the class. |
| **Closure** | 1. Summarize the lesson's learning outcomes and get affirmation from students on these.<br>2. Suggested Reading:<br>- "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 150-190.<br>- "Computer System Architecture" by M. Morris Mano, Chapter 5, pp. 200-245. |
| **Evaluation** | 1. How would you define register transfer in your own words?<br>2. What are the basic components involved in register transfer?<br>3. Why are registers essential in computer systems? |

| Lesson Plan No. 3 | Course Name: Computer Organization and architecture<br>Topic: Bus and Memory Transfer | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of a bus in computer architecture.<br>b. Describe the types of buses and their functions.<br>c. Explain memory transfer processes and mechanisms.<br>d. Identify the significance of bus and memory transfer in computer systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>- What do you understand by the term "bus" in computer architecture?<br>- How do you think data is transferred between different components of a computer system?<br>- Explain the importance of buses and memory transfer in computer performance and efficiency.<br><br>**2. Development (30 minutes)**<br>**a. What is a Bus?**<br>**-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.<br>**-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.<br><br>**b. Types of Buses:**<br>-Data Bus: Transfers actual data between the processor, memory, and peripheral devices.<br>-Address Bus: Carries the address location of where the data is to be read or written. It is unidirectional.<br>-Control Bus: Transfers control signals from the CPU to other components, coordinating data transfers and ensuring proper communication.<br>**c. Memory Transfer:**<br>-Definition: Memory transfer refers to the process of moving data between the main memory and the processor or between different memory locations.<br>-Mechanisms: |

| | |
|---|---|
| | -Direct Memory Access (DMA): Allows peripheral devices to directly transfer data to/from memory without CPU intervention, increasing efficiency.<br>-Memory-Mapped I/O: Peripheral devices are assigned specific memory addresses, enabling the CPU to read/write data as if it were memory.<br>**d. Importance of Bus and Memory Transfer:**<br>-System Performance: Efficient bus and memory transfer mechanisms are essential for high-performance computing, as they reduce bottlenecks and ensure smooth data flow.<br>-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.<br>**e. YouTube Video:**<br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the different types of buses and their roles in memory transfer. Discuss the answers with the class. |
| **Closure** | 1. Summarize the lesson learning outcomes and ensure students understand the key concepts.<br>2. Suggested Reading:<br>- "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 5, pp. 200-235.<br>- "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 3, pp. 100-145. |
| **Evaluation** | 1. How would you define a bus in computer architecture in your own words?<br>2. What are the different types of buses and their functions?<br>3. Why is memory transfer essential in computer systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 4 | Course Name: Computer Organization and architecture<br>Topic: Arithmetic Micro-Operations | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept and significance of arithmetic micro-operations in computer architecture.<br>b. Identify and describe different types of arithmetic micro-operations.<br>c. Explain how arithmetic micro-operations are implemented in digital systems.<br>d. Apply arithmetic micro-operations in practical scenarios. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>- What do you understand by the term "arithmetic micro-operations"?<br>- Can you give examples of basic arithmetic operations you use in daily calculations?<br>- Explain the relevance of arithmetic micro-operations in computer systems.<br><br>**2. Development (30 minutes)**<br>**a. What are Arithmetic Micro-Operations?**<br>-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.<br>-Importance: Discuss how arithmetic micro-operations facilitate various computational tasks, including addition, subtraction, increment, decrement, and shift operations, enabling complex arithmetic calculations in processors.<br><br>**b. Types of Arithmetic Micro-Operations:**<br>-Addition: Binary addition of two register contents, generating a sum and carry.<br>-Subtraction: Achieved by adding the complement of the subtrahend to the minuend.<br>-Increment: Increasing the binary value of a register by one.<br>-Decrement: Decreasing the binary value of a register by one.<br>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. |

| | |
|---|---|
| | **c. Implementation in Digital Systems:**<br>-Binary Adder-Subtractor: A circuit that can perform both addition and subtraction based on control signals.<br>-Arithmetic Logic Unit (ALU): The component of the CPU where arithmetic micro-operations are executed.<br><br>**d. Applications of Arithmetic Micro-Operations:**<br>-Real-World Applications: Discuss examples such as mathematical computations in processors, data manipulation in memory, and algorithms in various software applications.<br>-Case Study: Walk through the design and implementation of a simple arithmetic circuit using binary adder and subtractor components.<br><br>**e. YouTube Video:**<br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the types of arithmetic micro-operations and their functions.<br>- Discuss the answers with the class. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading:<br>- "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 85-120.<br>- "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 4, pp. 150-190. |
| **Evaluation** | 1. How would you define arithmetic micro-operations in your own words?<br>2. What are the different types of arithmetic micro-operations?<br>3. How are arithmetic micro-operations implemented in digital systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 5 | Course Name: Computer Networks<br>Topic: Instruction Cycle | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concept of the instruction cycle.<br>b. Identify the different phases of the instruction cycle.<br>c. Explain the significance of each phase in the execution of instructions.<br>d. Recognize the applications of the instruction cycle in modern computer systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br><br>● What do you understand by the term "instruction cycle"?<br>● How often do you think the instruction cycle is repeated in a computer's operation?<br>● Explain the relevance of understanding the instruction cycle in today's computing technology.<br><br>**Development (30 minutes)**<br><br>**What is an Instruction Cycle?**<br><br>● **Definition:**<br>○ 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.<br>● **Importance:**<br>○ 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.<br><br>**Phases of the Instruction Cycle:**<br><br>● **Fetch Phase:**<br>○ **Definition:**<br>■ The CPU fetches the instruction from memory. This phase involves reading the instruction stored at the address pointed to by the program counter (PC).<br>○ **Components:**<br>■ Program Counter (PC): Holds the address of the next instruction to be fetched.<br>■ Memory Address Register (MAR): Holds the address of the memory location to be read.<br>■ 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** | • Summarize the lesson learning outcomes and get affirmation from students on these.<br><br>Suggested Reading<br><br>• "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 123-145.<br>• "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 3, pp. 87-112. |
| **Evaluation** | • How would you define the instruction cycle in your own words?<br>• What are the basic phases of the instruction cycle?<br>• Why is the instruction cycle essential in computer systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 6 | Course Name: Computer Organization and architecture<br>Topic: Shift Micro Operation | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of shift micro operations.<br>b. Identify different types of shift operations.<br>c. Explain the significance of shift operations in computer architecture.<br>d. Demonstrate how shift operations are used in various applications. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>- Begin with a brief discussion: "What do you understand by the term 'shift operation' in computer architecture?"<br>- Ask: "Can you think of any instances where shifting data might be useful?"<br>- Explain the relevance of shift operations in data processing and manipulation within computer systems.<br><br>**2. Development (30 minutes)**<br>**a. What are Shift Micro Operations?**<br>-Definition: Shift micro operations are used to shift the bits of a binary number to the left or right.<br>-These operations are fundamental in various arithmetic and logical operations in a computer system.<br>-Importance: Shifting bits is crucial for multiplication and division operations.<br>They help in aligning data and manipulating bit patterns for efficient computation.<br>**b. Types of Shift Operations:**<br>**-Logical Shift:** Shifts bits to the left or right, with zeros filling the vacated positions.<br>Example: Logical left shift of 1010 becomes 0100.<br>**-Arithmetic Shift:** Similar to logical shift but maintains the sign bit (for signed numbers).<br>Example: Arithmetic right shift of 1101 becomes 1110.<br>**-Circular Shift (Rotate):** Bits shifted out from one end are reintroduced at the other end.<br>Example: Circular left shift of 1011 becomes 0111 (and the leftmost bit is reintroduced at the rightmost position).<br>**c. Applications of Shift Operations:** Multiplication and Division: Shifting left by one position is equivalent to |

| | |
|---|---|
| | multiplying by 2; shifting right by one position is equivalent to dividing by 2.<br>-Data Packing and Unpacking: Efficiently manage data storage by packing multiple smaller data units into a larger one or unpacking them.<br>-Bitwise Operations: Used in various algorithms, encryption, and encoding techniques.<br>**d. Practical Examples and Demonstration:**<br>-Show examples of shift operations in binary numbers using slides.<br>-Demonstrate a simple circuit design using basic shift registers.<br>-Discuss a case study of a real-world application, such as data encryption, where shift operations are used.<br>**e. YouTube Video:**<br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the types of shift operations and their specific use cases.<br>- Discuss the answers with the class. |
| **Closure** | 1. Summarize the lesson learning outcomes and confirm students' understanding.<br>2. Suggested Reading:<br>- "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 200-240.<br>- "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 5, pp. 150-180. |
| **Evaluation** | 1. How would you define shift micro operations in your own words?<br>2. What are the different types of shift operations and their significance?<br>3. Why are shift operations essential in computer systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 1 | Course Name: Computer Networks<br>Topic: Instruction Code | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of instruction codes.<br>b. Identify different types of instruction formats.<br>c. Explain the significance of opcode and operand in an instruction.<br>d. Recognize the role of instruction codes in CPU operations.<br>e. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br><br>• Start with a question: "What do you understand by the term 'instruction code'?"<br>• Discuss how instructions are essential for a computer to perform tasks.<br>• Briefly explain the relevance of instruction codes in the CPU's operation and overall system functionality.<br><br>**Development (30 minutes)**<br><br>**What are Instruction Codes?**<br><br>• 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).<br>• Importance: Explain how instruction codes enable the CPU to perform various tasks like data manipulation, control operations, and input/output operations.<br><br>**Basic Components of Instruction Codes:**<br><br>• **Opcode (Operation Code):**<br>○ The part of the instruction that specifies the operation to be performed (e.g., add, subtract, load, store).<br>• **Operand:**<br>○ The part of the instruction that specifies the data or the address of the data on which the operation is to be performed.<br><br>**Instruction Formats:**<br><br>• **Discuss different instruction formats like:**<br>○ Zero Address Instructions: (e.g., stack-based operations)<br>○ One Address Instructions: (e.g., accumulator-based operations)<br>○ Two Address Instructions: (e.g., direct memory operations)<br>○ Three Address Instructions: (e.g., register operations) |

| | |
|---|---|
| | **Role of Instruction Codes in CPU Operations:**<br><br>• Explain how the CPU fetches, decodes, and executes instructions.<br>• Discuss the instruction cycle and how instruction codes are crucial for its execution.<br><br>**Examples and Case Studies:**<br><br>• Provide examples of simple instruction codes and how they are used in basic CPU operations.<br>• Walk through a case study of a simple program and explain how the instruction codes are executed.<br><br>**YouTube Video:**<br><br>**Exercise (5 minutes)**<br><br>• Ask students to write down an example of an instruction code and explain its components.<br>• Discuss the examples with the class. |
| **Closure** | • Summarize the lesson by reiterating the key points about instruction codes, their components, and their importance in CPU operations.<br>• Confirm that students understand the lesson's learning outcomes.<br><br>**Suggested Reading**<br><br>• "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 45-85.<br>• "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 2, pp. 60-90. |
| **Evaluation** | • How would you define an instruction code in your own words?<br>• What are the basic components of an instruction code?<br>• Why are opcodes essential in the instruction cycle?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 2 | Course Name: Computer Networks<br>Topic: Computer Register | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of computer registers.<br>b. Identify different types of registers and their functions.<br>c. Explain the significance of registers in CPU operations.<br>d. Recognize the applications of registers in modern computer systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br><br>• What do you understand by the term "computer register"?<br>• How do you think registers are used in the CPU?<br>• Explain the relevance of registers in performing CPU operations.<br><br>**Development (30 minutes)**<br><br>**What are Computer Registers?**<br><br>• **Definition:**<br>○ Explain computer registers as small, fast storage locations within the CPU used to hold data and instructions temporarily.<br>• **Importance:**<br>○ Discuss how registers enable quick data access and manipulation during CPU operations. Highlight their role in executing instructions efficiently.<br><br>**Types of Computer Registers:**<br><br>• **General-Purpose Registers:**<br>○ Used for general data storage and arithmetic operations. Examples include AX, BX, CX, and DX in x86 architecture.<br>• **Special-Purpose Registers:**<br>○ Used for specific functions within the CPU. Examples include:<br>– Program Counter (PC): Holds the address of the next instruction to be executed.<br>– Instruction Register (IR): Holds the currently executing instruction.<br>– Accumulator (ACC): Used in arithmetic and logic operations.<br>– Stack Pointer (SP): Points to the top of the stack in memory.<br>– Status Register (FLAGS): Holds condition codes and control bits. |

| | |
|---|---|
| | **Functionality of Registers:**<br><br>• **Data Transfer:**<br>  ○ Registers facilitate data transfer between the CPU and memory or I/O devices.<br>• **Arithmetic and Logic Operations:**<br>  ○ Registers store operands for arithmetic and logic operations performed by the ALU.<br>• **Control Operations:**<br>  ○ Registers control the flow of instructions and manage execution states.<br><br>**Applications of Registers in Modern Computer Systems:**<br><br>• **Real-World Applications:**<br>  ○ Discuss examples such as the use of registers in executing machine-level instructions in processors.<br>  ○ Highlight the role of registers in handling interrupts and context switching in operating systems.<br>• **Case Study:**<br>  ○ Walk through the use of registers in a simple assembly language program.<br><br>**YouTube Video:**<br><br>**Exercise (5 minutes)**<br>• Ask students to summarize the types of registers and their functions.<br>• Discuss the answers with the class. |
| **Closure** | • Summarize the lesson learning outcomes and get affirmation from students on these.<br>**Suggested Reading**<br>• "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 85-120.<br>• "Computer Architecture: A Quantitative Approach" by John L. Hennessy and David A. Patterson, Chapter 2, pp. 50-80. |
| **Evaluation** | • How would you define computer registers in your own words?<br>• What are the basic types of registers in the CPU?<br>• Why are registers essential in CPU operations?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 3 | Course Name: Computer Networks<br>Topic: Computer Instructions | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of computer instructions.<br>b. Identify different types of computer instructions and their formats.<br>c. Explain the significance of instruction sets in computer architecture. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br>• Start with a question: "What do you think happens when you give a command to a computer?"<br>• Discuss the basic idea that computers follow instructions to perform tasks.<br>• Explain the importance of understanding computer instructions in the field of computer science and engineering.<br><br>**Development (30 minutes)**<br><br>**What are Computer Instructions?**<br>• 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.<br>• Importance: Discuss how computer instructions are crucial for executing programs, controlling hardware, and processing data.<br><br>**Basic Components of Computer Instructions:**<br>• Instruction Format: Explain the structure of an instruction, typically consisting of an opcode (operation code) and operands (data or memory locations).<br>o Opcode: The part of an instruction that specifies the operation to be performed (e.g., ADD, SUB, LOAD).<br>o Operands: The data on which the operation is performed or the addresses where data is stored.<br><br>**Types of Computer Instructions:**<br>• Data Transfer Instructions: Instructions that move data between registers, memory, and I/O devices (e.g., MOV, LOAD, STORE).<br>• Arithmetic Instructions: Instructions that perform arithmetic operations (e.g., ADD, SUB, MUL, DIV).<br>• Logical Instructions: Instructions that perform logical operations (e.g., AND, OR, NOT). |

| | |
|---|---|
| | • Control Instructions: Instructions that change the sequence of execution (e.g., JMP, CALL, RET).<br>• Input/Output Instructions: Instructions that handle I/O operations (e.g., IN, OUT).<br><br>**Real-World Applications:**<br><br>• Discuss how different types of instructions are used in various applications, such as operating systems, application software, and embedded systems.<br>• Case Study: Walk through a simple program and identify the types of instructions used.<br><br>**YouTube Video:**<br><br>**Exercise (5 minutes)**<br><br>• Ask students to summarize the basic types of computer instructions and their functions.<br>• Discuss the answers with the class. |
| **Closure** | Summarize the lesson learning outcomes and get affirmation from students on their understanding.<br>Suggested Reading:<br>• "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 2, pp. 50-90.<br>• "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 3, pp. 100-130. |
| **Evaluation** | • How would you define computer instructions in your own words?<br>• What are the basic components of a computer instruction?<br>• Why are instruction sets essential in computer systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 4 | Course Name: Computer Networks<br>Topic: Timing and Control Signals | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of timing and control signals.<br>b. Identify different types of timing and control signals used in computer architecture.<br>c. Explain the significance of these signals in the execution of instructions. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br>• Question: What do you understand by the term "timing and control signals"?<br>• Discussion: How often do you think about the processes that occur within a computer when it executes an instruction?<br>• Relevance: Explain the importance of timing and control signals in ensuring that instructions are executed correctly and efficiently within a CPU.<br>**Development (30 minutes)**<br>1. **What are Timing and Control Signals?**<br>○ **Definition:**<br>■ Timing signals coordinate the timing of various operations within the CPU.<br>■ Control signals manage the actions of the CPU, ensuring that the correct operations are performed at the right time.<br>○ **Importance:**<br>■ Discuss how these signals are crucial for synchronizing the operations of various components within the CPU.<br>■ Highlight their role in the proper execution of instructions, data transfer, and control flow.<br>2. **Types of Timing and Control Signals:**<br>○ **Clock Signal:**<br>■ The clock signal provides a consistent timing reference for the sequential operations of the CPU.<br>○ **Control Signals:**<br>■ Fetch: Signals the retrieval of an instruction from memory.<br>■ Decode: Initiates the interpretation of the fetched instruction.<br>■ Execute: Triggers the execution of the instruction.<br>■ Memory Read/Write: Manages the reading from and writing to memory. |

| | |
|---|---|
| | ■ Interrupts: Signals that an external event requires immediate attention from the CPU.<br>3. **Significance of Timing and Control Signals:**<br>○ **Synchronization:**<br>■ Ensures all parts of the CPU work together harmoniously.<br>○ **Efficiency:**<br>■ Minimizes delays and maximizes throughput by coordinating operations.<br>○ **Reliability:**<br>■ Prevents errors by providing clear instructions for each operation.<br>4. **Applications in Modern Computer Systems:**<br>○ **Real-World Examples:**<br>■ Discuss how timing and control signals are used in modern CPUs, such as in multi-core processors and advanced computing systems.<br>○ **Case Study:**<br>■ Walk through the execution of a simple instruction cycle, illustrating the role of timing and control signals at each stage.<br>5. **YouTube Video:**<br><br>**Exercise (5 minutes)**<br><br>• Ask students to summarize the different types of timing and control signals and their significance.<br>• Discuss the answers with the class. |
| **Closure** | • Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>• Suggested Reading:<br>○ "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 120-145.<br>○ "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 3, pp. 85-112.<br><br>Homework:<br><br>• Write a short essay on the role of control signals in modern computer processors and upload it on the learning management system. |
| **Evaluation** | • How would you define timing and control signals in your own words?<br>• What are the basic types of control signals used in a CPU?<br>• Why are clock signals essential in digital systems?<br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 5 | Course Name: Computer Organization and architecture<br>Topic: Logic Micro Operations | Course No.:403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of logic micro operations.<br>b. Identify different types of logic micro operations.<br>c. Explain the significance of logic micro operations in computer systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>- What do you understand by the term "logic micro operations"?<br>- How often do you encounter logical operations in computing tasks?<br>- Explain the relevance of logic micro operations in the functioning of CPUs and ALUs.<br><br>**2. Development (30 minutes)**<br>**a. What are Logic Micro Operations?**<br>-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.<br>-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.<br>**b. Types of Logic Micro Operations:**<br>• **AND Operation:**<br>-Outputs true if both corresponding bits of the operands are true.<br>-Example: 1010 AND 1100 = 1000<br>• **OR Operation:**<br>-Outputs true if at least one corresponding bit of the operands is true.<br>-Example: 1010 OR 1100 = 1110<br>• **XOR Operation:**<br>-Outputs true if only one of the corresponding bits is true.<br>-Example: 1010 XOR 1100 = 0110<br>• **NOT Operation:**<br>-Inverts the bits of the operand.<br>-Example: NOT 1010 = 0101 |

| | |
|---|---|
| | **c. Truth Tables:**<br><br>-Provide truth tables for AND, OR, XOR, and NOT operations.<br>-Discuss the output for all possible input combinations.<br><br>**d. Applications of Logic Micro Operations:**<br>● **Real-World Applications:**<br>-Discuss examples like error detection and correction, data masking, and bitwise manipulation in various algorithms.<br>-Highlight the use of logic operations in encryption and compression algorithms.<br>● **Case Study:**<br>-Walk through the design of a simple logic circuit using basic logic operations.<br><br>**e. YouTube Video:**<br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the different types of logic micro operations and their functions.<br>- Discuss the answers with the class. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>- "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4, pp. 125-160.<br>- "Computer System Architecture" by M. Morris Mano, Chapter 5, pp. 200-225. |
| **Evaluation** | 1. How would you define logic micro operations in your own words?<br>2. What are the different types of logic micro operations?<br>3. Why are logic micro operations essential in digital systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 6 | Course Name: Computer Networks<br>Topic: Memory Reference Instructions | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of memory reference instructions.<br>b. Identify different types of memory reference instructions and their functions.<br>c. Explain the execution cycle of memory reference instructions.<br>d. Recognize the significance of memory reference instructions in computer architecture. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br><br>● Start with a question: "What do you understand by the term 'memory reference instruction'?"<br>● Discuss the relevance of memory reference instructions in computer systems and their role in program execution.<br><br>**Development (30 minutes)**<br><br>**Memory Reference Instructions:**<br><br>**1. Definition:**<br>○ Explain that memory reference instructions are commands in machine language that direct the processor to perform operations involving memory locations.<br>**2. Types of Memory Reference Instructions:**<br>○ Load (LDA): Transfers data from a memory location to a register.<br>○ Store (STA): Transfers data from a register to a memory location.<br>○ Add (ADD): Adds data from a memory location to a register.<br>○ Subtract (SUB): Subtracts data from a memory location to a register.<br>○ Jump (JMP): Changes the sequence of instruction execution by setting the program counter to a specified address.<br>**3. Execution Cycle:**<br>○ Fetch: The instruction is fetched from memory.<br>○ Decode: The control unit decodes the fetched instruction to determine the operation and the operands.<br>○ Execute: The specified operation is performed, and the result is stored as needed.<br>**4. Importance in Computer Architecture:**<br>○ Discuss how memory reference instructions are crucial for data manipulation and control flow in programs. |

| | |
|---|---|
| | ○ Highlight their role in implementing algorithms and enabling complex computations.<br><br>**Examples:**<br><br>● Provide example assembly code snippets to illustrate the use of memory reference instructions.<br>● Walk through the execution of each instruction step-by-step.<br><br>**Exercise (5 minutes)**<br>● Ask students to write down the sequence of operations for a given memory reference instruction.<br>● Discuss the answers with the class and provide feedback. |
| **Closure** | ● Summarize the key points covered in the lesson.<br>● Ensure students understand the different types of memory reference instructions and their execution cycles.<br><br>**Suggested Reading:**<br><br>● "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 4.<br>● "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 5. |
| **Evaluation** | ● How would you define memory reference instructions in your own words?<br>● What are the different types of memory reference instructions, and what do they do?<br>● Describe the execution cycle of a memory reference instruction.<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 7 | Course Name: Computer Networks<br>Topic: Input-Output Interrupts | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of interrupts in computer systems.<br>b. Identify different types of interrupts.<br>c. Explain the working of input-output interrupts.<br>d. Recognize the importance of interrupts in efficient computer operation. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br><br>1. Ask students: What do you understand by the term "interrupt" in computer systems?<br>2. Discuss real-life examples where interruptions are necessary and beneficial.<br>3. Explain the relevance of interrupts in managing multiple tasks efficiently in computers.<br><br>**Development (30 minutes)**<br><br>**1. What are Interrupts?**<br>○ Definition: Explain interrupts as signals that inform the CPU of an event that needs immediate attention, pausing the current CPU operations.<br>○ Importance: Discuss how interrupts improve the efficiency of CPU operations by allowing multitasking and handling external events.<br>**2. Types of Interrupts:**<br>○ Hardware Interrupts: Triggered by hardware devices (e.g., keyboard, mouse).<br>○ Software Interrupts: Generated by software instructions or programs.<br>○ Maskable Interrupts: Can be turned off by the CPU to avoid interruption.<br>○ Non-maskable Interrupts: Cannot be turned off and must be processed.<br>**3. Input-Output (I/O) Interrupts:**<br>○ **Functioning:**<br>■ Describe how I/O interrupts allow peripheral devices to communicate with the CPU.<br>■ 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.<br>○ **Example:** Explain the process with an example of a keyboard input interrupt. |

| | |
|---|---|
| | 4. **Advantages of Using Interrupts:**<br>○ Efficient CPU utilization by allowing multitasking.<br>○ Faster response to external events compared to polling.<br>○ Reduced CPU idle time.<br>5. **Case Study:**<br>○ Walk through a simple case where a printer uses interrupts to notify the CPU when it is ready to receive more data.<br>6. **YouTube Video:**<br><br>**Exercise (5 minutes)**<br>● Ask students to summarize the types of interrupts and their functions.<br>● Discuss their answers with the class to ensure understanding. |
| **Closure** | Summarize the lesson's learning outcomes and confirm that students understand the key concepts of input-output interrupts.<br>Suggested Reading:<br>● "Computer Organization and Architecture" by William Stallings, Chapter 7, pp. 240-275.<br>● "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 4, pp. 150-190. |
| **Evaluation** | ● How would you define an interrupt in your own words?<br>● What are the basic types of interrupts?<br>● Why are input-output interrupts essential for computer operation?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 8 | Course Name: Computer Networks<br>Topic: Design of Basic Computer | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic design and components of a computer.<br>b. Identify the main functional units of a computer and their roles.<br>c. Explain the operation and interconnection of these units. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br><br>● Question to the class: What comes to your mind when you think of a computer?<br>● Discussion: How do you think a computer processes information?<br>● Relevance: Explain the importance of understanding computer design for anyone working with technology.<br><br>**Development (30 minutes)**<br><br>**1. Overview of Basic Computer Design:**<br>○ **Definition:**<br>■ A basic computer is an electronic device that processes data according to a set of instructions (program).<br>■ Comprises various components working together to perform operations.<br>**2. Main Components of a Basic Computer:**<br>○ **Central Processing Unit (CPU):**<br>■ Control Unit (CU): Directs operations of the computer.<br>■ Arithmetic Logic Unit (ALU): Performs arithmetic and logical operations.<br>○ **Memory Unit:**<br>■ Primary Memory (RAM): Temporary storage for data and instructions.<br>■ Secondary Memory: Permanent storage (e.g., hard drives, SSDs).<br>○ **Input and Output Devices:**<br>■ Input Devices: Devices like keyboard, mouse, etc., for data entry.<br>■ Output Devices: Devices like monitors, printers, etc., for data output.<br>**3. Interconnection of Components:**<br>○ **Buses: Communication pathways connecting CPU, memory, and I/O devices.**<br>■ Data Bus: Transfers data.<br>■ Address Bus: Transfers information about where data should be sent or retrieved.<br>■ Control Bus: Transfers control signals. |

| | |
|---|---|
| | 4. **Instruction Set Architecture (ISA):**<br>○ Definition: The set of instructions a computer can execute.<br>○ Operation: How the CPU interprets and executes these instructions.<br>○ Examples: Machine language instructions for data movement, arithmetic operations, and control operations.<br>5. **Basic Operations of a Computer:**<br>○ Fetch: Retrieving an instruction from memory.<br>○ Decode: Interpreting the instruction.<br>○ Execute: Performing the operation specified by the instruction.<br>○ Store: Writing the result back to memory if needed.<br>6. **Case Study:**<br>○ **Simple Computer Operation:**<br>■ Example of a simple program execution cycle using the basic operations (fetch, decode, execute, store). |
| **Closure** | ● Summary: Recap the lesson learning outcomes.<br>● Confirmation: Get affirmation from students on their understanding of the key concepts.<br>Suggested Reading<br>● "Computer Organization and Architecture" by William Stallings, Chapter 1, pp. 1-40.<br>● "Computer Systems: A Programmer's Perspective" by Randal E. Bryant and David R. O'Hallaron, Chapter 2, pp. 50-85. |
| **Evaluation** | ● How would you describe the role of the CPU in a computer?<br>● What are the key components of a computer's memory system?<br>● Why is the instruction set architecture important in computer design?<br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 9 | Course Name: Computer Networks<br>Topic: Design of Accumulator Logic | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of an accumulator in a computer system.<br>b. Identify the role and function of accumulator logic in processing operations.<br>c. Design a simple accumulator logic circuit.<br>d. Apply accumulator logic in basic arithmetic operations. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **Introduction (5 minutes)**<br><br>● Begin with a brief discussion on what an accumulator is.<br>○ "Who has heard the term 'accumulator' in the context of computer systems before?"<br>○ "How do you think an accumulator functions in the execution of programs?"<br>● Explain the significance of accumulator logic in the CPU and its role in executing instructions efficiently.<br><br>**Development (30 minutes)**<br><br>**1. What is an Accumulator?**<br>○ **Definition:**<br>■ Describe an accumulator as a register in a CPU where intermediate arithmetic and logic results are stored.<br>■ Highlight its role in reducing the time required to fetch data from memory.<br>○ **Importance:**<br>■ Discuss how an accumulator simplifies the architecture and increases the speed of the CPU by handling intermediate results directly.<br>■ Explain its use in various arithmetic operations such as addition, subtraction, and logical operations.<br>**2. Design of Accumulator Logic:**<br>○ **Basic Structure:**<br>■ Illustrate the basic structure of an accumulator with a diagram.<br>■ Components include input lines, output lines, and control signals.<br>○ **Operation:**<br>■ Explain the operation of an accumulator with simple examples:<br>■ Adding two numbers.<br>■ Storing the result. |

| | |
|---|---|
| | ■ Retrieving and modifying the result for further operations.<br>○ **Logic Circuit Design:**<br>■ Step-by-step design of a simple accumulator circuit.<br>■ Show how logic gates (AND, OR, NOT) and flip-flops are used to construct the accumulator.<br>■ Explain the control logic needed to manage data flow into and out of the accumulator.<br>**3. Applications of Accumulator Logic:**<br>○ **Real-World Examples:**<br>■ Discuss the use of accumulators in simple processors and microcontrollers.<br>■ Highlight the role of accumulators in signal processing and embedded systems.<br>○ **Case Study:**<br>■ Walk through the design and simulation of an accumulator used in a basic CPU model.<br><br>**YouTube Video:**<br><br>**Exercise (5 minutes)**<br>● Ask students to design a simple accumulator circuit using logic gates and flip-flops.<br>● Discuss their designs and clarify any doubts. |
| **Closure** | ● Summarize the lesson's key points: the role of accumulators, their design, and their applications.<br>● Confirm students' understanding through a quick Q&A session.<br>Suggested Reading<br>● "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 3, pp. 85-110.<br>● "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 4, pp. 130-150. |
| **Evaluation** | ● How would you define an accumulator in your own words?<br>● What are the primary components of accumulator logic?<br>● Why is accumulator logic essential in a CPU?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 1 | Course Name: Computer Organization and Architecture<br>Topic: Processor organization | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the fundamental concepts of processor organization.<br>b. Identify the components and their functions within a processor.<br>c. Analyze the relationship between processor organization and computer architecture. |
| **Teaching Aids (if any)** | a. Projector, Slides<br>b. Use of Nearpod tool for online quiz |
| **Teaching Development** | 1) **Introduction** (5 minutes)<br>a. Ask questions to gauge prior knowledge:<br>b. What is processor organization?<br>c. Can you name any components of a processor?<br>d. Discuss the importance of processor organization in computer architecture.<br><br>2) **Development** (30 minutes)<br>a. Explain the basic components of processor organization:<br>b. Arithmetic Logic Unit (ALU), Control Unit, Registers, Memory Interface.<br>c. Discuss the role of each component in instruction execution and data processing.<br>**3) Exercise (5 minutes)**<br>a. Binary to Decimal: Convert the binary number 101101101101 to its decimal equivalent.<br>b. Decimal to Hexadecimal: Convert the decimal number 9292 to its hexadecimal representation.<br>c. Octal to Binary: Convert the octal number 345345 to binary form. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>"Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy.<br>3. Spend 5 minutes to wrap up and consolidate the learnings |
| **Evaluation** | 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA<br>Spend 5 minutes to evaluate student assimilation of the lesson contents |

| Lesson Plan No. 2 | Course Name: Computer Organization and Architecture<br>Topic: General-purpose registers | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of general register organization.<br>b. Identify the purpose and functions of general-purpose registers. |
| **Teaching Aids (if any)** | a. Projector, Slides<br>b. Use of Nearpod tool for online quiz |
| **Teaching Development** | **1) Introduction (5 minutes)**<br>a. What are registers in a computer system?<br>b. Why are registers important in computer organization?<br>c. Define general register organization and its significance.<br>d. Discuss the importance of general-purpose registers in processor operations.<br>e. Provide examples of general-purpose registers in typical processor architectures.<br><br>2) **Development** (30 minutes)<br>a. Explain the concept of registers and their organization within a processor.<br>b. Discuss the purpose of general-purpose registers, special-purpose registers, and program counter (PC).<br>c. Introduce RTL as a symbolic language used to describe register transfer operations.<br>d. Provide examples of RTL statements to demonstrate register operations.<br><br>**3) Exercise (5 minutes)**<br>a. Why are registers important in computer architecture?<br>b. How do registers contribute to efficient instruction execution? |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>"Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy.<br>3. Spend 5 minutes to wrap up and consolidate the learnings |
| **Evaluation** | 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA<br>Spend 5 minutes to evaluate student assimilation of the lesson contents |

| Lesson Plan No. 3 | Course Name: Computer Organization and Architecture<br>Topic: Stack organization | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of stack organization.<br>b. Explain the role of a stack in computer architecture.<br>c. Learn about stack operations and their implementations. |
| **Teaching Aids (if any)** | a. Projector, Slides<br>b. Use of Nearpod tool for online quiz |
| **Teaching Development** | **1) Introduction (5 minutes)**<br>a. Ask questions:<br>b. What is stack organization?<br>c. Why is stack organization important in computer architecture?<br>d. How does a stack work in programming languages?<br>e. Discuss the basic concept of stack organization and its significance.<br>f. Introduce the concept of Last In First Out (LIFO) data structure.<br><br>2) **Development** (30 minutes)<br>a. Basic Stack Operations:<br>b. Push: Adding an item to the top of the stack.<br>c. Pop: Removing an item from the top of the stack.<br>d. Discuss how stacks are implemented in computer memory.<br><br>**3) Exercise (5 minutes)**<br>a. Stack Operations:<br>b. Perform push and pop operations on a stack with given elements. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>"Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy.<br>3. Spend 5 minutes to wrap up and consolidate the learnings |
| **Evaluation** | 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA<br>Spend 5 minutes to evaluate student assimilation of the lesson contents |

| Lesson Plan No. 4 | Course Name: Computer Organization and Architecture<br>Topic: Addressing modes | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Define addressing modes in computer architecture<br>b. Understand different types of addressing modes<br>c. Analyze the use cases for various addressing modes. |
| **Teaching Aids (if any)** | d. Projector, Slides<br>e. Use of Nearpod tool for online quiz |
| **Teaching Development** | **1) Introduction (5 minutes)**<br>a. Start with questions to engage students:<br>b. What is an addressing mode?<br>c. Why are addressing modes important in computer architecture?<br>d. Define addressing mode and its significance.<br><br>2) **Development** (30 minutes)<br>a. Basic Concepts of Addressing Modes:<br>b. Direct Addressing Mode<br>c. Indirect Addressing Mode<br>d. Register Addressing Mode<br>e. Immediate Addressing Mode<br>f. Indexed Addressing Mode<br>g. Relative Addressing Mode<br><br>**3) Exercise (5 minutes)**<br>a. Discuss scenarios for each addressing mode and ask students to identify the most suitable mode for each scenario. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>3. " Computer System Architecture," by Morris Mano.<br>4. Spend 5 minutes to wrap up and consolidate the learnings |
| **Evaluation** | 1. Reflective Questions (What, why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA<br>Spend 5 minutes to evaluate student assimilation of the lesson contents |

| Lesson Plan No. 5 | Course Name: Computer Organization and Architecture<br>Topic: Booth's algorithm | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the principles of binary multiplication.<br>b. Comprehend Booth's algorithm and its significance in computer arithmetic.<br>c. Implement Booth's algorithm for binary multiplication. |
| **Teaching Aids (if any)** | a. Projector, Slides<br>b. Use of Nearpod tool for online quiz |
| **Teaching Development** | **1. Introduction** (5 minutes)<br>• Ask questions to gauge students' understanding of binary multiplication.<br>• Define binary multiplication and its importance in computer arithmetic.<br>• Briefly introduce Booth's algorithm and its historical context.<br>**2. Development** (30 minutes)<br>a. Binary Multiplication Basics:<br>• Explain the principles of binary multiplication.<br>• Illustrate with simple examples.<br>b. Mathematical Background:<br>• Have a basic understanding of mathematical concepts<br>• particularly in arithmetic, as many topics involve mathematical operations.<br>c. Booth's Algorithm:<br>• Introduce Booth's algorithm, including its purpose and benefits.<br>• Explain the algorithm step-by-step with a clear example.<br>• Highlight the handling of positive and negative multipliers.<br>**3. Exercise** (5 minutes)<br>• Provide a binary multiplication problem for students to solve using Booth's algorithm.<br>• Encourage students to explain each step of the algorithm as they solve the problem. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>**Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy**<br>(This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.)<br>3. **"Digital Design" by M. Morris Mano and Michael D. Ciletti**<br>4. Spend 5 minutes to wrap up and consolidate the learnings |
| **Evaluation** | 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA |

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

| Lesson Plan No. 6 | Course Name: Computer Organization and Architecture<br>Topic: Division Algorithms | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the principles of division algorithms, including restoring and non-restoring methods.<br>b. Comprehend the step-by-step process involved in both restoring and non-restoring division. |
| **Teaching Aids (if any)** | c. Projector, Slides<br>d. Use of Nearpod tool for online quiz |
| **Teaching Development** | **1. Introduction** (5 minutes)<br>• Ask questions to assess prior knowledge about basic arithmetic operations.<br>• Define division in the context of binary arithmetic.<br>• Introduce the concept of division algorithms and differentiate between restoring and non-restoring division.<br>**2. Development** (30 minutes)<br>• Understanding Division Algorithms:<br>• Explain the basic concept of division in binary numbers.<br>• Define and explain the steps involved in the restoring division algorithm.<br>• Illustrate with a detailed example on how restoring division works.<br>• Define and explain the steps involved in the non-restoring division algorithm.<br>• Illustrate with a detailed example on how non-restoring division works.<br>**3. Exercise** (5 minutes)<br>• Describe the process and significance of quotient and remainder in division algorithms.<br>• Outline the steps in both restoring and non-restoring division algorithms with a simple binary example. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>**Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy**<br>(This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.)<br>3. **"Digital Design" by M. Morris Mano and Michael D. Ciletti**<br>4. Additional online resources and video tutorials on division algorithms in computer architecture. |
| **Evaluation** | 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA |

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

| Lesson Plan No. 7 | Course Name: Computer Organization and Architecture<br>Topic: Floating Point Representation | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the representation of floating-point numbers.<br>b. Comprehend the arithmetic operations involving floating point numbers, including addition, subtraction, multiplication, and division. |
| **Teaching Aids (if any)** | c. Projector, Slides<br>d. Use of Nearpod tool for online quiz |
| **Teaching Development** | **1. Introduction** (5 minutes)<br>• Ask questions to gauge the students' prior knowledge about floating point numbers.<br>• Define floating point numbers and explain their significance in computer systems.<br>• Introduce the IEEE 754 standard for floating point arithmetic.<br>**2. Development** (30 minutes)<br>**a. Understanding Floating Point Representation:**<br>• Explain the components of a floating-point number: sign, exponent, and mantissa (or significand).<br>• Describe the normalization of floating-point numbers.<br>**b. Floating Point Arithmetic Operations:**<br>• Addition and Subtraction:<br>• Aligning exponents, adding/subtracting significands, normalizing the result.<br>• Multiplication and Division: Multiplying/dividing significands, adding/subtracting exponents, normalizing the result.<br>• Demonstrate examples of each operation step-by-step.<br>**3. Exercise** (5 minutes)<br>• Provide a set of floating-point numbers and ask students to manually perform addition and multiplication operations. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>**Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy**<br>(This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.)<br>3. **"Digital Design" by M. Morris Mano and Michael D. Ciletti**<br>4. Spend 5 minutes to wrap up and consolidate the learnings. |
| **Evaluation** | 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA |

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

| Lesson Plan No. 8 | Course Name: Computer Organization and Architecture<br>Topic: Arithmetic operations | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the principles of decimal arithmetic operations in computer systems.<br>b. Comprehend the implementation of decimal addition, subtraction, multiplication, and division in a CPU |
| **Teaching Aids (if any)** | c. Projector, Slides<br>d. Use of Nearpod tool for online quiz |
| **Teaching Development** | **1. Introduction** (5 minutes)<br>• Ask introductory questions about arithmetic operations.<br>• Define decimal arithmetic and its significance in computer systems.<br>**2. Development** (30 minutes)<br>• Discuss the importance of decimal arithmetic in computing.<br>• Introduce the concepts of BCD (Binary-Coded Decimal) representation and its use in decimal arithmetic operations.<br>• Explain the process of performing decimal addition and subtraction.<br>• Illustrate with examples the step-by-step procedure of adding and subtracting decimal numbers using BCD.<br>**3. Exercise** (5 minutes)<br>• Basic Concepts of Decimal Arithmetic:<br>• Describe how decimal arithmetic is handled in modern processors.<br>• Outline the steps for performing a BCD addition and identify potential challenges. |
| **Closure** | 1. Summarize the Lesson Learning Outcomes and get affirmation from students on these.<br>2. Suggested Reading<br>**Computer Organization and Design: The Hardware/Software Interface" by David A. Patterson and John L. Hennessy**<br>(This book provides a comprehensive introduction to computer organization and architecture, covering topics such as instruction sets, memory hierarchy, and processor design.)<br>3. **"Digital Design" by M. Morris Mano and Michael D. Ciletti**<br>4. Spend 5 minutes to wrap up and consolidate the learnings. |
| **Evaluation** | 1. Reflective Questions (What, Why, Who?). Allow students to answer and discuss.<br>2. Nearpod Quiz on COA<br><br>Spend 5 minutes to evaluate student assimilation of the lesson contents |

| Lesson Plan No. 1 | Course Name: Computer Organization and architecture<br>Topic: Input/Output Interface | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of Input/Output (I/O) Interface.<br>b. Identify and describe different types of I/O interfaces and their functions.<br>c. Explain the role of I/O interfaces in computer systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– Start by asking: "How do computers interact with external devices like keyboards, monitors, and printers?"<br>– 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.<br>– Explain the importance of I/O interfaces in ensuring efficient communication between the computer and peripheral devices.<br><br>**2. Development (30 minutes)**<br>**a. Overview of Input/Output Interface**<br>**-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.<br><br>**b. Basic Components of an I/O Interface:**<br>**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.<br>**Data Registers:** Explain how data registers within the I/O interface temporarily hold data being transferred between the CPU and peripherals.<br><br>**c. Types of I/O Interfaces**<br>● **Programmed I/O:**<br>Discuss how the CPU actively monitors the status of the I/O device and controls the data transfer process.<br><br>● **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.<br>● **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.<br>**3. Exercise (5 minutes)**<br>- Ask students to list the types of I/O interfaces discussed and briefly explain their functions.<br>- Discuss their answers to ensure clarity and understanding.. |
| **Closure** | 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic.<br>2. Suggested Reading<br>- "Computer System Architecture" by M. Morris Mano, Chapter 7, pp. 230-275.<br>- "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 6, pp. 260-300. |
| **Evaluation** | 1. How would you define an I/O interface in your own words?<br>2. What are the different types of I/O interfaces, and how do they differ in operation?<br>3. Why are I/O interfaces essential in computer systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 2 | Course Name: Computer Organization and architecture<br>Topic: Asynchronous Data Transfer | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of asynchronous data transfer.<br>b. Identify and describe the methods and protocols used in asynchronous data transfer.<br>c. Explain the significance of asynchronous data transfer in computer systems. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– Begin by asking: "What do you understand by the term 'Asynchronous Data Transfer'?"<br>– Discuss how asynchronous data transfer differs from synchronous data transfer.<br>– Explain the relevance of asynchronous data transfer in handling communication between devices that operate at different speeds.<br><br>**2. Development (30 minutes)**<br>**a. Overview of asynchronous data transfer**<br>**-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.<br><br>**b. Importance of Asynchronous Data Transfer:**<br>Discuss how asynchronous data transfer allows devices with different clock rates to communicate effectively.<br>Highlight its role in serial communication protocols such as UART (Universal Asynchronous Receiver/Transmitter).<br><br>**c. Applications of Asynchronous Data Transfer:**<br>● **Real-World Applications:**<br>Discuss examples such as RS-232 serial communication and USB (Universal Serial Bus) interfaces.<br><br>Highlight the use of asynchronous transfer in modem communications and other telecommunication systems.<br><br>**3. Exercise (5 minutes)**<br>– Ask students to summarize the key components of asynchronous data transfer and their functions. |

| | |
|---|---|
| | - 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.<br>2. Suggested Reading<br>- "Computer Networks" by Andrew S. Tanenbaum, Chapter 2, pp. 45-90.<br>- "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 6, pp. 275-320. |
| **Evaluation** | 1. How would you define asynchronous data transfer in your own words?<br>2. What are the key components of asynchronous data transfer?<br>3. Why is asynchronous data transfer essential in digital communication systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 3 | Course Name: Computer Organization and architecture<br>Topic: Mode of Data Transfer | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concepts of different modes of data transfer.<br>b. Identify and describe various methods used for transferring data between devices.<br>c. Explain the significance of each mode in terms of speed, efficiency, and application. |
| **Teaching Aids (if any)** | d. Presentation slides.<br>e. Projector<br>f. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– Begin by asking: "How do computers transfer data between devices, such as memory and I/O devices?"<br>– Discuss the importance of data transfer in computer systems and how it affects the performance of the system.<br>– Explain how different data transfer modes are suited for different tasks.<br><br>**2. Development (30 minutes)**<br>**a. Overview of Data Transfer**<br>**-Definition:** Define data transfer as the movement of data between two or more components in a computer system.<br><br>**b. Basic Components of Data Transfer:**<br>● **Programmed I/O:**<br>Discuss how the CPU actively monitors the status of the I/O device and controls the data transfer process.<br><br>● **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.<br>● **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.<br>**3. Exercise (5 minutes)**<br>– Ask students to summarize the key differences between the various modes of data transfer.<br>– 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.<br>2. Suggested Reading |

| | |
|---|---|
| | - "Computer Organization and Design" by David A. Patterson and John L. Hennessy, Chapter 7, pp. 320-350.<br>- "Structured Computer Organization" by Andrew S. Tanenbaum, Chapter 4, pp. 250-290. |
| **Evaluation** | 1. How would you describe the different modes of data transfer in your own words?<br>2. What are the advantages of using DMA over programmed I/O?<br>3. Why is interrupt-driven I/O preferred in real-time systems?<br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 4 | Course Name: Computer Organization and architecture<br>Topic:Priority Interrupt | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of priority interrupt and its significance in computer systems.<br>b. Identify different types of interrupts and their priority levels.<br>c. Explain how priority interrupt mechanisms are implemented in hardware. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– "What happens when multiple devices need the CPU's attention simultaneously?"<br>– Discuss the concept of interrupts and why prioritizing them is essential for system efficiency.<br>– Introduce the term "priority interrupt" and its relevance in computer architecture.<br><br>**2. Development (30 minutes)**<br>**a. What is Priority Interrupt?**<br>**-Definition:** Explain that a priority interrupt is a system that manages multiple interrupts by assigning them different priority levels.<br>Highlight that the CPU responds to the highest priority interrupt first, ensuring critical tasks are handled promptly.<br><br>**b. Types of Interrupts:**<br>● External Interrupts<br>● Internal Interrupts<br>● Software Interrupts<br><br>**c. Importance of Priority Interrupt:**<br>-Discuss the role of priority interrupt in ensuring system reliability and efficiency.<br><br>-Highlight how it prevents low-priority tasks from blocking high-priority ones, improving overall system performance..<br><br>**3. Exercise (5 minutes)**<br>– Ask students to summarize the key components of asynchronous data transfer and their functions.<br>– 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.<br>2. Suggested Reading<br>- "Computer Networks" by Andrew S. Tanenbaum, Chapter 2, pp. 45-90.<br>- "Digital Design and Computer Architecture" by David Harris and Sarah Harris, Chapter 6, pp. 275-320. |
| **Evaluation** | 1. How would you define asynchronous data transfer in your own words?<br>2. What are the key components of asynchronous data transfer?<br>3. Why is asynchronous data transfer essential in digital communication systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 5 | Course Name: Computer Organization and architecture<br>Topic:Direct Memory Access (DMA) | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concept of Direct Memory Access (DMA).<br>b. Identify and describe the role of DMA in data transfer within computer systems.<br>c. Explain how DMA improves system performance by bypassing the CPU during data transfer. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– What are the traditional methods of data transfer between memory and peripherals?"<br>– Discuss the limitations of CPU-driven data transfer (e.g., Programmed I/O and Interrupt-driven I/O).<br>– Introduce DMA as a method that allows peripherals to communicate directly with memory, bypassing the CPU.<br><br>**2. Development (30 minutes)**<br>**a. What is Direct Memory Access?**<br>**-Definition:** Explain DMA as a feature that allows peripherals to directly read from and write to the main memory, reducing the CPU's workload.<br><br>**b. Basic Components of DMA:**<br>● DMA Controller<br>● DMA Channels<br>● Software Interrupts<br><br>**c. Importance of Priority Interrupt:**<br>-Performance Improvement:Discuss how DMA increases system efficiency by allowing data transfer without burdening the CPU.<br><br>Highlight the reduced latency in data transfer, especially in high-speed applications.<br><br>**3. Exercise (5 minutes)**<br>- Ask students to summarize the advantages of using DMA in system design.<br>- 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. |

| | |
|---|---|
| | 2. Suggested Reading<br>- "Computer Organization and Design" by David Patterson and John Hennessy, Chapter 6, pp. 300-350.<br>- "Modern Computer Architecture" by John Paul Shen, Chapter 7, pp. 150-200. |
| **Evaluation** | 1. How does DMA improve system performance compared to traditional data transfer methods?<br>2. What are the key components of a DMA system?<br>3. Explain a real-world application where DMA is crucial.<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 6 | Course Name: Computer Organization and architecture<br>Topic: Input-Output Processor (IOP) | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concept of an Input-Output Processor (IOP).<br>b. Explain the need for an IOP in a computer system.<br>c. Describe the functioning and structure of an IOP.<br>d. Recognize the difference between an IOP and a central processing unit (CPU). |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– Begin by asking: "What do you understand by the term 'Input-Output Processor'?"<br>– Discuss the role of input-output operations in computer systems.<br>– Explain the necessity of offloading input-output tasks from the CPU to an IOP for better efficiency and performance.<br><br>**2. Development (30 minutes)**<br>**a. Definition of Input-Output Processor (IOP)**<br>**-Definition:**Explain that the IOP is a specialized processor designed to handle input-output tasks, reducing the load on the CPU.<br><br>**b. Basic Components of an Input-Output Processor:**<br>● Control Unit<br>● Registers<br>● Interface Logic<br><br>**c. Functions of the IOP:**<br>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.<br><br>**3. Exercise (5 minutes)**.<br>- Ask students to summarize the role of an Input-Output Processor and its components.<br>- Engage in a brief discussion to ensure the students have understood the concepts. |
| **Closure** | 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic. |

| | |
|---|---|
| | 2. Suggested Reading<br>- "Computer Organization and Design" by David Patterson and John Hennessy, Chapter 6, pp. 300-350.<br>- "Modern Computer Architecture" by John Paul Shen, Chapter 7, pp. 150-200. |
| **Evaluation** | 1. How would you define an Input-Output Processor in your own words?<br>2. What are the basic functions of an IOP?<br>3. Why is an IOP beneficial in high-performance systems?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 7 | Course Name: Computer Organization and architecture<br>Topic: Memory Hierarchy | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept and significance of memory hierarchy in a computer system.<br>b. Identify and describe the different levels of memory hierarchy (registers, cache, main memory, secondary storage).<br>c. Explain the characteristics and performance trade-offs at each level of the memory hierarchy. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– "Why do we have different types of memory in a computer system?"<br>– Introduce the concept of memory hierarchy, highlighting the need for different levels of memory to balance speed and cost.<br>– Explain that memory hierarchy is a structured way to manage data storage and access efficiently within a computer.<br><br>**2. Development (30 minutes)**<br>**a. Definition of Memory Hierarchy**<br>**-Definition:**Memory hierarchy is a layered structure that categorizes memory into levels based on speed, cost, and size.<br><br>**b. Levels of Memory Hierarchy**<br>● Registers<br>● Cache Memory<br>● Main Memory (RAM)<br>● Secondary Storage (Hard Drives, SSDs)<br><br>**c. Characteristics and Performance at Each Level**<br>Compare the characteristics of memory at each level in terms of access time, size, cost, and purpose.<br><br>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.<br><br>**3. Exercise (5 minutes)**.<br>– Ask students to summarize the different levels of memory hierarchy and their key characteristics.<br>– 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.<br>2. Suggested Reading<br>- "Computer Organization and Design" by David Patterson and John Hennessy.<br>- "Modern Computer Architecture" by John Paul Shen. |
| **Evaluation** | 1. How would you explain the concept of memory hierarchy?<br>2. What are the key differences between cache memory and main memory?<br>3. Why is the memory hierarchy essential for CPU performance?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 8 | Course Name: Computer Organization and architecture<br>Topic: Cache Memory | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the basic concept of cache memory and its significance in computer architecture.<br>b. Explain the different types of cache memory and their functionality.<br>c. Recognize the importance of cache mapping techniques and cache performance measures |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– What is cache memory, and why do we need it in computer systems?"<br>– Briefly explain how cache memory is used to reduce the time to access data from the main memory.<br>– Highlight its importance in speeding up the performance of a computer system.<br><br>**2. Development (30 minutes)**<br>**a. Definition of Cache Memory**<br>**-Definition:**Explain cache memory as a smaller, faster type of volatile computer memory that provides high-speed data access to the processor.<br><br>**b. Types of Cache Memory**<br>● **L1 Cache**: Located within the processor itself; it is the smallest and fastest type of cache.<br>● **L2 Cache**: Larger than L1 and may be located on the CPU chip or close to it.<br>● **L3 Cache**: Larger and slower than L1 and L2 but still faster than main memory, shared across cores in multi-core processors.<br><br>**c. Cache Mapping Techniques:**<br>**Direct Mapping**: Each block of main memory maps to exactly one cache line.<br><br>**Associative Mapping**: Any block of main memory can map to any line of cache. |

| | |
|---|---|
| | **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.<br><br>**3. Exercise (5 minutes)**.<br>- Ask students to summarize the different types of cache and their functions.<br>- Discuss their responses and clarify any misconceptions. |
| **Closure** | 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic.<br>2. Suggested Reading<br>- "Computer Organization and Design" by David Patterson and John Hennessy.<br>- "Modern Computer Architecture" by John Paul Shen. |
| **Evaluation** | 1. Define cache memory in your own words.<br>2. What are the different types of cache memory?<br>3. How do cache mapping techniques affect cache performance?<br>4. How can cache memory improve the overall system performance?<br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 9 | Course Name: Computer Organization and architecture<br>Topic: Virtual Memory | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the concept of virtual memory and its significance in computer systems.<br>b. Explain how virtual memory manages memory hierarchy and improves performance.<br>c. Describe the process of paging and segmentation in virtual memory.<br>d. Recognize the role of the Memory Management Unit (MMU) in translating virtual addresses to physical addresses. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– Begin by asking: "Have you ever wondered how computers run programs larger than their physical memory?"<br>– Explain the concept of memory hierarchy and introduce virtual memory as a technique to extend physical memory using disk space.<br>– Discuss the relevance of virtual memory in modern operating systems.<br><br>**2. Development (30 minutes)**<br>**a. Definition of Virtual Memory**<br>**-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.<br><br>**b. Key Components of Virtual Memory:**<br>● **Paging**<br>● **Segmentation**<br>● **Memory Management Unit (MMU)**<br>● **Page Table**<br><br>**c. Benefits of Virtual Memory**<br>**Increased Multitasking**: More programs can run simultaneously. |

| | |
|---|---|
| | **Security and Isolation:** Each process has its own virtual memory, preventing interference.<br><br>3. **Exercise (5 minutes)**.<br>- Ask students to summarize the different types of cache and their functions.<br>- Discuss their responses and clarify any misconceptions. |
| **Closure** | 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic.<br>2. Suggested Reading<br>- "Computer Organization and Design" by David Patterson and John Hennessy.<br>- "Modern Computer Architecture" by John Paul Shen. |
| **Evaluation** | 1. How would you define virtual memory in your own words?<br>2. What is the purpose of the Memory Management Unit (MMU)?<br>3. Can you explain the process of handling a page fault?<br>Spend 5 minutes to wrap up and consolidate the learnings. |

| Lesson Plan No. 10 | Course Name: Computer Organization and architecture<br>Topic:Memory Management System | Course No.: COM-403 |
|---|---|---|

| | |
|---|---|
| **Objectives** | At the end of the lesson the student shall be able to:<br>a. Understand the fundamental concepts of memory management in operating systems.<br>b. Identify and describe different memory allocation techniques such as paging, segmentation, and virtual memory.<br>c. Explain the role of memory management in resource allocation and process management. |
| **Teaching Aids (if any)** | a. Presentation slides.<br>b. Projector<br>c. You Tube video |
| **Teaching Development** | **1. Introduction (5 minutes)**<br>– What do you think happens to the memory when you run multiple programs on your computer?"<br>– Discuss the importance of memory management in ensuring that the operating system efficiently allocates and deallocates memory to different processes.<br>– Introduce the concept of memory management as a critical component of the operating system that oversees the usage of RAM and virtual memory.<br><br>**2. Development (30 minutes)**<br>**a. Definition of Memory Management**<br>-Define memory management as the process of coordinating and controlling computer memory, assigning portions to various running programs to optimize overall system performance.<br><br>**b. Memory Allocation Techniques**<br>● **Contiguous Memory Allocation**<br>● **Segmentation**<br>● **Paging**<br>● **Paging**<br><br>**c. Role of Memory Management in Resource Allocation**<br>-Discuss how memory management ensures that different processes get the memory they need while avoiding conflicts through techniques like swapping and dynamic loading.<br>-Highlight the concept of memory protection, ensuring that one process does not affect the memory allocated to another process.<br><br>**3. Exercise (5 minutes)**. |

| | |
|---|---|
| | - Ask students to identify and explain the difference between paging and segmentation.<br>- Discuss the answers with the class to ensure comprehension of the memory management techniques. |
| **Closure** | 1. Summarize the lesson learning outcomes and ask students for their feedback on the topic.<br>2. Suggested Reading<br>- "Operating System Concepts" by Abraham Silberschatz, Peter Baer Galvin, and Greg Gagne, Chapter 8, pp. 300-350.<br>- "Modern Operating Systems" by Andrew S. Tanenbaum, Chapter 3, pp. 120-170. |
| **Evaluation** | 1. How would you describe memory management in operating systems?<br>2. What is the difference between paging and segmentation?<br>3. Why is virtual memory important for running large applications?<br><br>Spend 5 minutes to wrap up and consolidate the learnings. |