**week 2 summary:**

**Topics and Key Learnings**

**Lecture 2-1: Conditional and Iterative Processing (Part I)**

1. **Conditional Logic:**

• **If Statements:** Syntax and examples to evaluate conditions and control program flow.

• **If-Else Statements:** Decision-making logic for alternate paths.

• **Nested Conditions:** Handling multiple conditions with nested if-else structures.

• **Conditional Operator:** A shorthand for if-else statements (condition ? expression1 : expression2).

2. **Iterative Logic:**

• **While Loops:**

• Basic syntax and examples, including edge-case considerations.

• Example: Printing numbers from 0 to .

• **String Processing Using Loops:**

• Example: Removing vowels from a string.

3. **Basic Examples and Patterns:**

• **Palindrome Checking:** Using while loops and string methods to check if a string is a palindrome.

• **Printing Patterns:**

• Examples include squares of asterisks and pyramids.

**Lecture 2-2: Conditional and Iterative Processing (Part II)**

1. **For Loops:**

• Iterating with counters and calculating values.

• Examples:

• Printing integers from 0 to .

• Calculating powers of two.

• Summing numbers and computing factorials.

2. **Enhanced Problem Solving with Loops:**

• Checking palindromes with for loops.

• Nested loops to create multiplication tables.

3. **Advanced Iterative Techniques:**

• **Do-While Loops:**

• Example: Generating random points inside a unit circle using the rejection method.

• Difference between while and do-while loops (execution of the loop body at least once in do-while).

4. **Numeral Systems:**

• Conversion between binary and decimal:

• Binary to Decimal using loops and powers of 2.

• Decimal to Binary through integer division and modulus.

5. **Monte Carlo Simulations:**

• **Gambler’s Ruin Problem:**

• Simulating the probability of reaching a goal starting with an initial amount.

• Use of randomization to model and compute statistical probabilities.

**Themes Across Lectures 2-1 and 2-2**

• Mastery of conditional constructs (if, if-else, nested conditions).

• Use of iterative structures (while, for, do-while) for problem-solving.

• Practical applications in numerical conversions, string processing, and statistical simulations.

• Emphasis on algorithmic thinking and debugging edge cases.