# Lab #1: Install Go Lang Programming

- Step 1: Download and Install Go
- Step 2: Verify Installation
- Step 3: Set Up the Environment Variables
- Step 4: Writing Your First Go Program ( Hello World!)
- Step 5: Run the Program
- Step 6: Build and Execute

- **Task 1:** Write a Go program to demonstrate the use of assignment operators. The program should take two integer inputs and perform various assignment operations such as =, +=, -=, *=, /=, and %=. Display the result after each operation.

# Lab #2: Logical Operators

- Write a Go program to demonstrate the use of logical operators such as &&, ||, and !.
- The program should take two integer inputs and evaluate logical expressions like:
  - **both** positive (&&)
  - one greater than the other (||)
  - not equal (!)

  Display whether each condition is true or false.

# Lab #3: Bitwise and Assignment Opr.

- Write a Go program to demonstrate the use of bitwise and assignment operators. The program should perform AND, OR, XOR, NOT, left shift, and right shift operations on two integers using functions and display the results. Also, show the effect of assignment operators on variable values.
- Pseudocode:

```
main() {
        myXor(a,b)
        myNOT(a,b)
        myOR(a,b)
        myAND(a,b)
}
```

# Lab #4: Mini Calculator

- Write a Go program to implement a mini calculator using functions. The program should read two numbers and an operator from the user, perform the operation, and return/print the result. Support at least the operators: +, -, *, /, %. Handle division by zero and invalid operators gracefully. (Optional) Make it menu-driven and repeat until the user chooses Exit.

- ===== Mini Calculator =====

  1) Add  2) Sub  3) Mul  4) Div  5) Mod  6) Exit

  Choose: 4

  Enter a: 10

  Enter b: 0

  Error: division by zero

# **Lab #5:** Binary, Hex, and Base64 Encoding

- Write a Go program to demonstrate how text data can be represented in binary, hexadecimal, and Base64 formats.
- The program should take a string input and display its equivalent binary, hexadecimal, and Base64 representations.

# **Lab #6:** XOR Encryption/Decryption

- Write a Go program to perform simple XOR encryption and decryption.
- The program should take a plaintext message and a key (single byte or repeating key) and produce a ciphertext by XORing them.
- Use the same function to decrypt the ciphertext to recover the original message.
  - func xorEncrypt(text string, key byte) string {  Code here }