**Slide-3:**

; Assembly program using XCHG, ADD, SUB, INC, DEC, and NEG with output display

; Note: MOV and INT instructions are used for setup and display

.model small

.stack 100h

.data

val1 DB 7 ; First value

val2 DB 4 ; Second value

result DB 0 ; For storing result

msg DB "Result: $" ; Message string

.code

main PROC

; Setup data segment (required)

MOV AX, @data

MOV DS, AX

; Load values into registers

MOV AL, val1 ; AL = 7

MOV BL, val2 ; BL = 4

MOV CL, 0 ; CL = 0

; Core logic using only XCHG, ADD, SUB, INC, DEC, and NEG

; Exchange AL and BL

XCHG AL, BL ; AL = 4, BL = 7

; Add BL to AL

ADD AL, BL ; AL = 4 + 7 = 11

; Increment AL

INC AL ; AL = 11 + 1 = 12

; Subtract BL from AL

SUB AL, BL ; AL = 12 - 7 = 5

; Increment AL again

INC AL ; AL = 5 + 1 = 6

; Decrement AL

DEC AL ; AL = 6 - 1 = 5

; Negate AL

NEG AL ; AL = -5

; Negate AL again

NEG AL ; AL = 5

; Store result

MOV result, AL

; Display message

MOV AH, 09h

LEA DX, msg

INT 21h

; Display result (for single digit)

MOV DL, result

ADD DL, 30h ; Convert to ASCII

MOV AH, 02h

INT 21h

; Exit program

MOV AH, 4Ch

INT 21h

main ENDP

END main

**Slide-4**

**Code-1**

.MODEL small

.STACK 100h

.DATA

msg db "Hello world $"

.CODE

MAIN PROC

mov ax,@DATA

mov ds,ax

keepGoing:

mov ah,09h

lea dx,msg

int 21h

mov ah,02h

mov dx,0Dh

int 21h

mov dx,0Ah

int 21h

jmp keepGoing

MAIN ENDP

END MAIN

**Code-2**

.MODEL small

.STACK 100h

.DATA

msg1 db "Hello world $"

msg2 db "bye world $"

.CODE

MAIN PROC

mov bx,1h

mov ax,@DATA

mov ds,ax

checkCondition:

cmp bx,5h

jg stop

jl keepGoing

keepGoing:

mov ah,09h

lea dx,msg1

int 21h

mov ah,02h

mov dx,0Dh

int 21h

mov dx,0Ah

int 21h

inc bx

jmp checkCondition

stop:

mov ah,09h

lea dx,msg2

int 21h

jmp end

end:

mov ah,4Ch

int 21h

MAIN ENDP

END MAIN