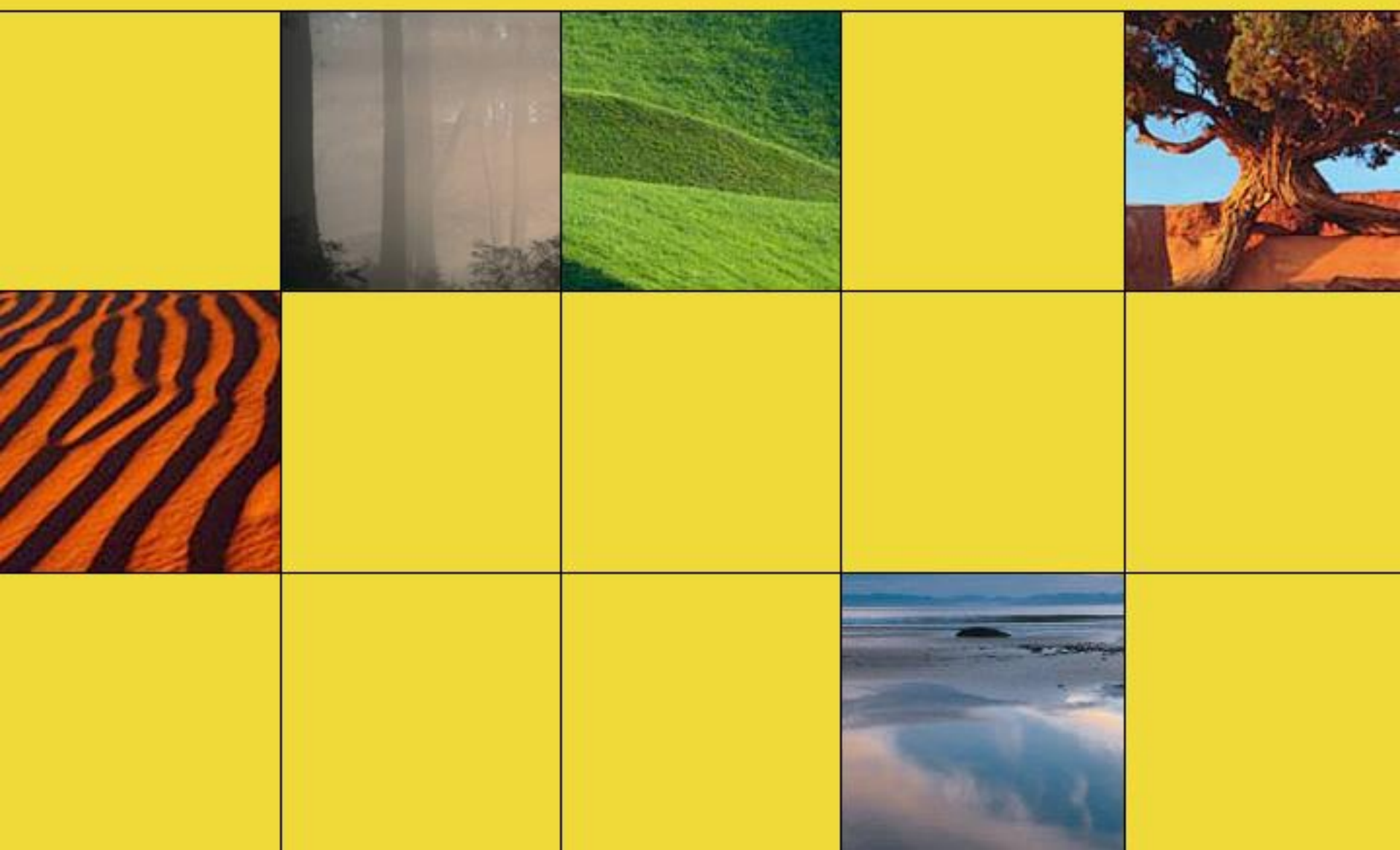
Fundamentals of Assembly language



Lesson Plan

- Learning Assembly language from examples

Instruction Execution and Addressing

- Executing an instruction include
  - Fetch the next instruction
  - Decode the instruction
  - Execute the instruction

Instruction Execution and Addressing

- Instruction address = Code Segment address (CS) + Instruction Offset (IP)
- Data address = Data Segment address (DS) + Data Offset

CS 26AE IP 0044

Instruction address = ???????

CS | 26AE | IP | 0044

Instruction address = 26AE0
+ 0044

26B24

DS | 25BD

26B24 → **A03F00**

Data address=??????

CS | 26AE | IP | 0044

Data address = 25BD0

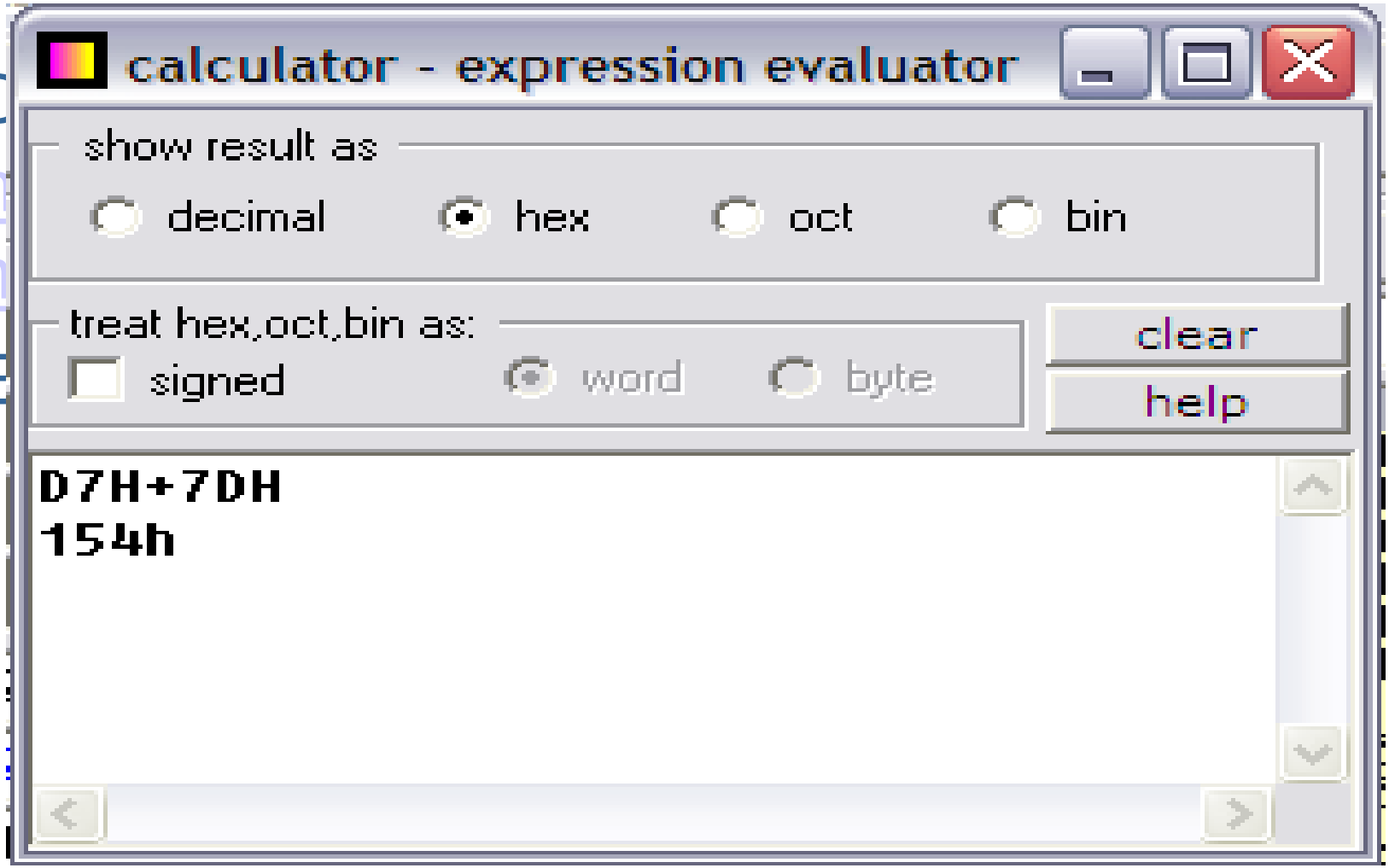
\+ 003F

25C0F

DS | 25BD

Tools

- D
  h
  h
- E


calculator - expression evaluator
show result as
decimal
hex
oct
bin
treat hex,oct,bin as:
signed
word
byte
clear
help
D7H+7DH
154h

Review old concepts

- Address of BIOS data area: starts at 0040H
- Boot process:
  - CS: FFFF (Segment address: FFFF0)
  - IP:0000

Example program

```
; Add two numbers and store the results into the third variable
page    60,132
TITLE          A04ASM1 (EXE)  Move and add operations
; -----------------------------------------------
STACK          SEGMENT PARA STACK 'Stack'
       DW        32 DUP(0)
STACK          ENDS
; -----------------------------------------------
DATASEG    SEGMENT PARA 'Data'
       FLDD                    DW              215
       FLDE                    DW              125
       FLDF                    DW              ?
DATASEG    ENDS
; -----------------------------------------------
CODESEG    SEGMENT PARA 'Code'
MAIN                           PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG                      ;Set address of data
                MOV     DS,AX                           ;  segment in DS
                MOV             AX,FLDD                 ;Move 0215 to AX
                ADD             AX,FLDE                 ;Add  0125 to AX
                MOV             FLDF,AX                 ;Store sum in FLDF
                MOV             AX,4C00H                ;End processing
                INT             21H
MAIN                           ENDP                                    ;End of procedure
CODESEG    ENDS                                        ;End of segment
                END             MAIN                    ;End of program
```

COMMENTS

; Add two numbers and store the results into the third variable

```
page    60,132
TITLE           A04ASM1 (EXE)  Move and add operations
; ---------------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
      DW          32 DUP(0)
STACK           ENDS
; ---------------------------------------------
DATASEG         SEGMENT PARA 'Data'
      FLDD                  DW          215
      FLDE                  DW          125
      FLDF                  DW          ?
DATASEG         ENDS
; ---------------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                    PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG          ;Set address of data
                MOV     DS,AX               ;Segment in DS
                MOV       AX,FLDD           ;Move 0215 to AX
                ADD       AX,FLDE           ;Add  0125 to AX
                MOV       FLDF,AX           ;Store sum in FLDF
                MOV       AX,4C00H          ;End processing
                INT       21H
MAIN                    ENDP                ;End of procedure
CODESEG         ENDS                        ;End of segment
END   MAIN                                  ;End of program
```

Comments

; <your comments>

IDENTIFIERS

```
; Add two numbers and store the results into the third variable
page    60,132
TITLE           A04ASM1 (EXE)  Move and add operations
; -----------------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
     DW           32 DUP(0)
STACK           ENDS
; -----------------------------------------------
DATASEG         SEGMENT PARA 'Data'
    FLDD            DW      0215H
    FLDE            DW      0125H
    FLDF            DW      ?
DATASEG         ENDS
; -----------------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG              ;Set address of data
                MOV     DS,AX                   ;Segment in DS
                MOV         AX,FLDD             ;Move 0215 to AX
                ADD         AX,FLDE             ;Add  0125 to AX
                MOV         FLDF,AX             ;Store sum in FLDF
                MOV         AX,4C00H            ;End processing
                INT         21H
MAIN                    ENDP                    ;End of procedure
```

Identifiers

- Identifier is a name applied to an item in a program to reference
  - Name (e.g: FLDD DW 215)
  - Label (e.g: MAIN PROC FAR)
- Identifiers must not a reserved word and only contain:
  - Alphabetic letters (A-Z,a-z)
  - Digits (0-9)
  - ?,_,$,@,dot (.) (but not for the first character)
- Maximum length is 247

Practice

- Change the variables in the existing program
- Assign new values to them
- Compile and run

RESERVED WORDS

- Instructions: ADD, MOV
- Directives: .TITLE, .MODEL
- Operators: FAR, SIZE
- Pre-defined symbols: @Data, @Model
- Register: AX,BX

STATEMENT

- Instructions: are translated to object code

  MOV, ADD, LEA..

- Directives: tell the assembler to perform a specific action.

[identifier] operation [operand(s)] [;comments]

STATEMENTS

```
; Add two numbers and store the results into the third variable
page    60,132
TITLE           A04ASM1 (EXE)  Move and add operations
; -----------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
      DW          32 DUP(0)
STACK           ENDS
; -----------------------------------------
DATASEG         SEGMENT PARA 'Data'
      FLDD                  DW          215
      FLDE                  DW          125
      FLDF                  DW          ?
DATASEG         ENDS
; -----------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                    PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG          ;Set address of data
                MOV     DS,AX               ;Segment in DS
                MOV         AX,FLDD         ;Move 0215 to AX
                ADD         AX,FLDE         ;Add  0125 to AX
                MOV         FLDF,AX         ;Store sum in FLDF
                MOV         AX,4C00H                    ;End processing
                INT         21H
MAIN                    ENDP                ;End of procedure
CODESEG         ENDS                        ;End of segment
END   MAIN                                  ;End of program
```

Directives

- Control the way a source program assembles and lists
- Generate no machine code (unlike instructions which generate object code)

Page directive

```
; Add two numbers and store the results into the third variable
page    60,132   page [length(10-255)],[width(60-132)]
TITLE           A04ASM1 (EXE)  Move and add operations
; ------------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
      DW          32 DUP(0)
STACK           ENDS
; ------------------------------------------
DATASEG         SEGMENT PARA 'Data'
      FLDD                  DW          215
      FLDE                  DW          125
      FLDF                  DW          ?
DATASEG         ENDS
; ------------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                          PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG              ;Set address of data
                MOV     DS,AX                   ;Segment in DS
                MOV       AX,FLDD               ;Move 0215 to AX
                ADD       AX,FLDE               ;Add  0125 to AX
                MOV       FLDF,AX               ;Store sum in FLDF
                MOV       AX,4C00H                        ;End processing
                INT       21H
MAIN                          ENDP                ;End of procedure
CODESEG         ENDS                              ;End of segment
```

Page directive

```
; Add two numbers and store the results into the third variable
page    10,70
TITLE           A04ASM1 (EXE)  Move and add operations
; -----------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
      DW          32 DUP(0)
STACK           ENDS
; -----------------------------------------
DATASEG         SEGMENT PARA 'Data'
      FLDD                  DW          215
      FLDE                  DW          125
      FLDF                  DW          ?
DATASEG         ENDS
; -----------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                    PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG          ;Set address of data
                MOV     DS,AX               ;Segment in DS
                MOV       AX,FLDD           ;Move 0215 to AX
                ADD       AX,FLDE           ;Add  0125 to AX
                MOV       FLDF,AX           ;Store sum in FLDF
                MOV       AX,4C00H                  ;End processing
                INT       21H
MAIN                    ENDP                ;End of procedure
CODESEG         ENDS                        ;End of segment
```

Title directive

```
; Add two numbers and store the results into the third variable
page    10,70
TITLE           A04ASM1 (EXE)  Move and add operations
; -----------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
     DW           32 DUP(0)
STACK           ENDS
; -----------------------------------------
DATASEG         SEGMENT PARA 'Data'
     FLDD                DW          215
     FLDE                DW          125
     FLDF                DW          ?
DATASEG         ENDS
; -----------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                    PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG          ;Set address of data
                MOV     DS,AX               ;Segment in DS
                MOV       AX,FLDD           ;Move 0215 to AX
                ADD       AX,FLDE           ;Add  0125 to AX
                MOV       FLDF,AX           ;Store sum in FLDF
                MOV       AX,4C00H                    ;End processing
                INT       21H
MAIN                    ENDP                ;End of procedure
CODESEG         ENDS                        ;End of segment
```

Segment directive

```
; Add two numbers and store the results into the third variable
page    60,132
TITLE           A04ASM1 (EXE)  Move and add operations
; -----------------------------------------
STACK       SEGMENT PARA STACK 'Stack'
    DW          32 DUP(0)
STACK       ENDS
; -----------------------------------------------
DATASEG   SEGMENT PARA 'Data'
    FLDD                DW          215
    FLDE                DW          125
    FLDF                DW          ?
DATASEG   ENDS
; -----------------------------------------------
CODESEG  SEGMENT PARA 'Code'
MAIN                    PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG              ;Set address of data
                MOV         FLDF,AX             ;Store sum in FLDF
                MOV         AX,4C00H                        ;End processing
                INT         21H
MAIN                    ENDP                    ;End of procedure
CODESEG     ENDS                                ;End of segment
END             MAIN                                        ;End of program
```

Segment directive

Name	Operation	Operand
Segment-name	SEGMENT	[align][combine] [`class’]
Segment-name	ENDS	

Example:

```
STACK   SEGMENT         PARA STACK 'Stack‘
STACK   ENDS
```

PROC directive

```
; Add two numbers and store the results into the third variable
page    60,132
TITLE           A04ASM1 (EXE)  Move and add operations
; -----------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
      DW          32 DUP(0)
STACK           ENDS
; -----------------------------------------
DATASEG         SEGMENT PARA 'Data'
      FLDD              DW          215
      FLDE              DW          125
      FLDF              DW          ?
DATASEG         ENDS
; -----------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG          ;Set address of data
                MOV     DS,AX               ;Segment in DS
                MOV       AX,FLDD           ;Move 0215 to AX
                MOV       FLDF,AX           ;Store sum in FLDF
                MOV       AX,4C00H                  ;End processing
                INT       21H
MAIN                ENDP            ;End of procedure
CODESEG         ENDS                ;End of segment
```

PROC directive

- Format:

  Procedure-name PROC Operand Comment

  Procedure-name ENDP

  Operand: relates to program execution (FAR)

ASSUME directive

```
; Add two numbers and store the results into the third variable
page    60,132
TITLE           A04ASM1 (EXE)  Move and add operations
; -----------------------------------------
STACK           SEGMENT PARA STACK 'Stack'
      DW          32 DUP(0)
STACK           ENDS
; -----------------------------------------
DATASEG         SEGMENT PARA 'Data'
      FLDD              DW          215
      FLDE              DW          125
      FLDF              DW          ?
DATASEG         ENDS
; -----------------------------------------
CODESEG         SEGMENT PARA 'Code'
MAIN                      PROC    FAR
                ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
                MOV     AX,DATASEG          ;Set address of data
                MOV     DS,AX               ;Segment in DS
                MOV       AX,FLDD           ;Move 0215 to AX
                MOV       FLDF,AX           ;Store sum in FLDF
                MOV       AX,4C00H                  ;End processing
                INT       21H

CODESEG         ENDS                        ;End of segment
END               MAIN                              ;End of program
```

ASSUME directive

- Tells the assembler the purpose of each segment in the program

Example:

```
ASSUME  SS:STACK,DS:DATASEG,CS:CODESEG
```

Simplified Segment Directives

- Model memory-model

	# Code segment	#Data segment
Small:	1, <=64K	1,<=64K
Medium:	any number,size	1, <=64K
Compact:	1, <=64K	any number,size
Large:	any number,size	any number,size
Huge:	any number,size	any number,size

Simplified Segment Directives

- STACK [size] (default: 1K)
- DATA (default size is 1K)
- CODE (default size is 1K)
- .EXIT directive

EQUATE directives

- Equal-Sign directive

  COEFFICIENT= 100

- EQU directive

  COEFFICIENT EQU 100

Data type

- Format for data definition

  [name] Dn expression

  Name: identifier

  Dn: Directives and can be:

DB: byte	DF:farword
DW: word	DQ:quadword
DD: doubleword	DT:tenbytes

  Expression:

  can be unnitialized: ?

  can be assigned a constant: such as 25, 21.

  Example:

  - DATAZ DB 21,22..
  - DW 10 DUP(?)

Data type

- Constant:
  - String: is defined within ‘ ‘ or “ “

  MESSAGE DB “I am learning assembly language”
  - Numeric:
    - Is stored in reverse sequence
    - Binary: 01B
    - Decimal: 9D(D is optional)
    - Hexadecimal: 1FH
    - Real: 12R

Directives for defining Data

- Byte: DB
  Word: DW

  Doubleword: DD

  Farword: DF

  Quadword: DQ

  Tenbytes: DT

Some instructions on arithmetic calculation

- ADD:
  ADD register register/memory/immediate
  Example: ADD AX,FLDE
- Subtract
  SUB register register/memory/immediate
  Example: SUB AX, 100
- Multiplication
  MUL register/memory
  Example: MUL CX
- Division
  DIV register/memory
  Example DIV CX