

## Assembly Language program

```
[org 0x0100]
; start of code

mov  ax, [value1]      ; move the constant 5 into register ax
mov  bx, [value2]      ; move the constant 10 into register bx

add  ax, bx            ; add value of bx into the value of ax

mov  bx, [value3]      ; add constant 15 into the value of bx
add  ax, bx            ; add value of bx into the value of ax
mov  [value4], ax      ; move the constant 0 into register ax

mov  ax, 0x4c00        ; exit ..
int  0x21              ; .. is what the OS should do for me

;Labels
value1: dw    5
value2: dw   10
value3: dw   15
value4: dw    0
```

## Listing for Object Code

```
1  [org 0x0100]
2  ; start of code
3
4
5  00000000 A1[1700]      mov  ax, [value1]
6  ; move the constant 5 into register ax
7
8  00000003 8B1E[1900]    mov  bx, [value2]
9  ; move the constant 10 into register bx
10
11 00000007 01D8          add  ax, bx
12 ; add value of bx into the value of ax
13
14 00000009 8B1E[1B00]    mov  bx, [value3]
15 ; add constant 15 into the value of bx
16
17 0000000D 01D8          add  ax, bx
18 ; add value of bx into the value of ax
19
20 0000000F A3[1D00]      mov  [value4], ax
21 ; move the constant 0 into register ax
22
23 00000012 B8004C        mov  ax, 0x4c00
24 ; exit ..
25
26 00000015 CD21          int  0x21
27 ; .. is what the OS should do for me
28
29                                ;Labels
30 00000017 0500          value1: dw    5
31 00000019 0A00          value2: dw   10
32 0000001B 0F00          value3: dw   15
33 0000001D 0000          value4: dw    0
34
35
36 ; watch the listing carefully
```

**Table: Status of Register, Memory and IP for Each Instruction**

S.NO.	INSTRUCTIONS	LABELS	MEMORY ADDRESS	MEMORY STATUS	REGISTER STATUS	IP STATUS	LISTING	MEMORY
1	mov ax, [value1]	value1: dw 5	0117	05 00	AX = 0005	0100	00000000 A1[1700]	3 Bytes
2	mov bx, [value2]	value2: dw 10	0119	0A 00	BX = 000A	0103	00000003 8B1E[1900]	4 Bytes
3	add ax, bx	-	-	-	AX = 000F	0107	00000007 01D8	2 Bytes
4	mov bx, [value3]	value3: dw 15	011B	0F 00	BX = 000F	0109	00000009 8B1E[1B00]	4 Bytes
5	add ax, bx	-	-	-	AX = 001E	010D	0000000D 01D8	2 Bytes
6	mov [value4], ax	value4: dw 0	011D	1E 00	AX = 001E	010F	0000000F A3[1D00]	3 Bytes
7	mov ax, 0x4c00	-	-	-	AX = 4C00	0112	00000012 B8004C	3 Bytes
8	int 0x21	-	-	-	AX = 0000	0115	00000015 CD21	2 Bytes