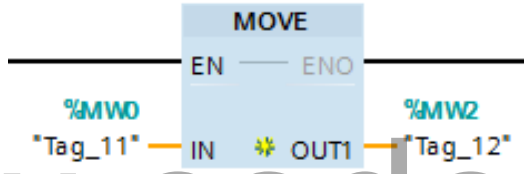
Siemens S7-1200

CPU 1212C AC/DC/Relay

Move Operations

- MOV
- MOV_BLK
- SWAP
- **Exercise Example**



Code and Compile
Learning Made Easy
www.codeandcompile.com

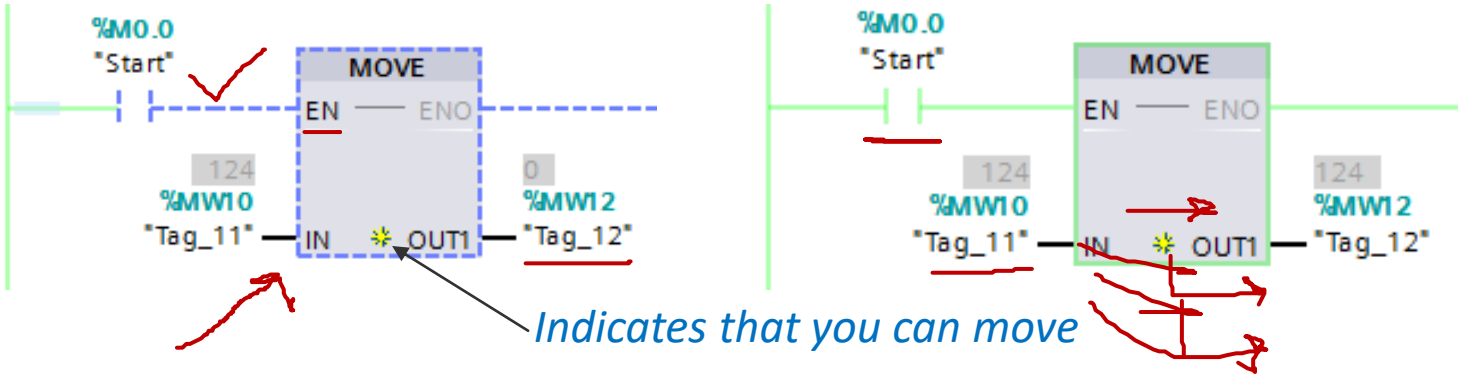
Understanding Move Operations - **MOVE**

Move Instructions

The **MOVE** instruction **copies** a single data element from the **source address** specified by the ***IN parameter*** to the **destination addresses** specified by the ***OUT parameter.***

Example:

Copying the value from MW10 to MW12



Indicates that you can move the input to multiple outputs

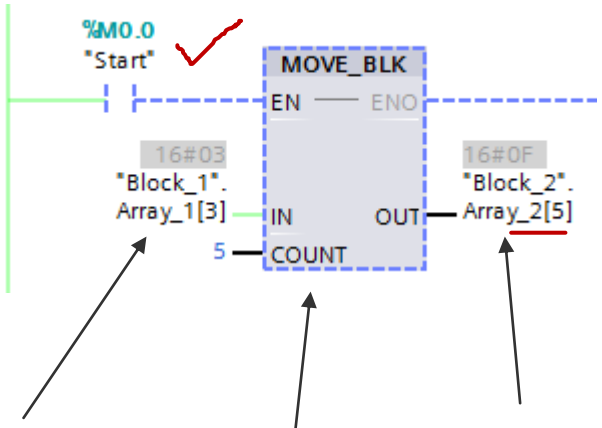
Understanding Move Operations – **MOVE_BLK**

Move Block Instruction

The **MOVE_BLK** instructions have an additional **COUNT** parameter. *The COUNT specifies how many data elements are copied.* The number of bytes per element copied depends on the data type assigned to the IN and OUT parameter tag names in the PLC tag table.

Example:

Copying the array element from source to destination



Input Array

Count

Output Array

	Name	Monitor value
1	Static	
2	Array_1	
3	Array_1[0]	16#00
4	Array_1[1]	16#01
5	Array_1[2]	16#02
6	Array_1[3]	16#03
7	Array_1[4]	16#04
8	Array_1[5]	16#05
9	Array_1[6]	16#06
10	Array_1[7]	16#07
11	Array_1[8]	16#08
12	Array_1[9]	16#09
13	Array_1[10]	16#0A
14	<Add new>	

	Name	Monitor value
1	Static	
2	Array_2	
3	Array_2[0]	16#0A
4	Array_2[1]	16#0B
5	Array_2[2]	16#0C
6	Array_2[3]	16#0D
7	Array_2[4]	16#0E
8	Array_2[5]	16#03
9	Array_2[6]	16#04
10	Array_2[7]	16#05
11	Array_2[8]	16#06
12	Array_2[9]	16#07
13	Array_2[10]	16#14

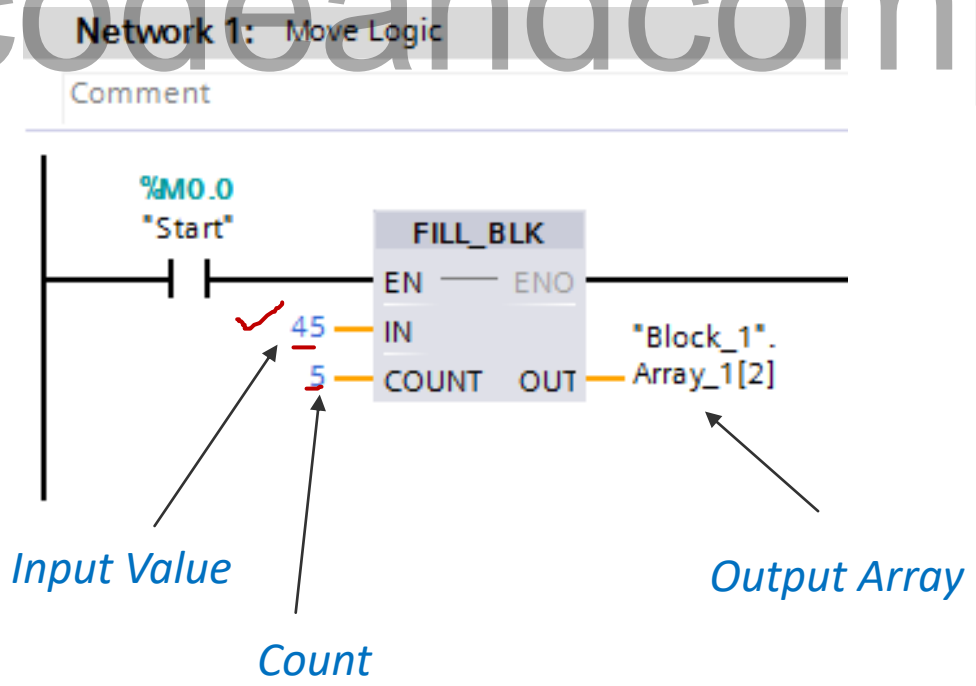
Understanding Move Operations – **FILL_BLK**

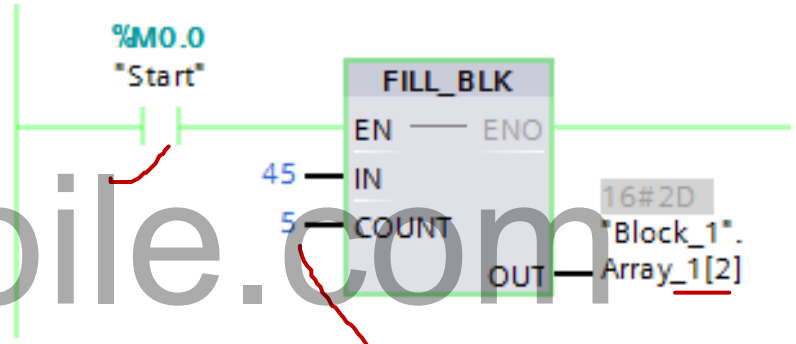
Fill Block Instruction

You can use the "Fill block" instruction to fill a memory area (destination area) with the value of the IN input. The destination area is filled beginning with the address specified at the OUT output.

Example:

Copying the constant value (45) to Block_1 Array_1[2]



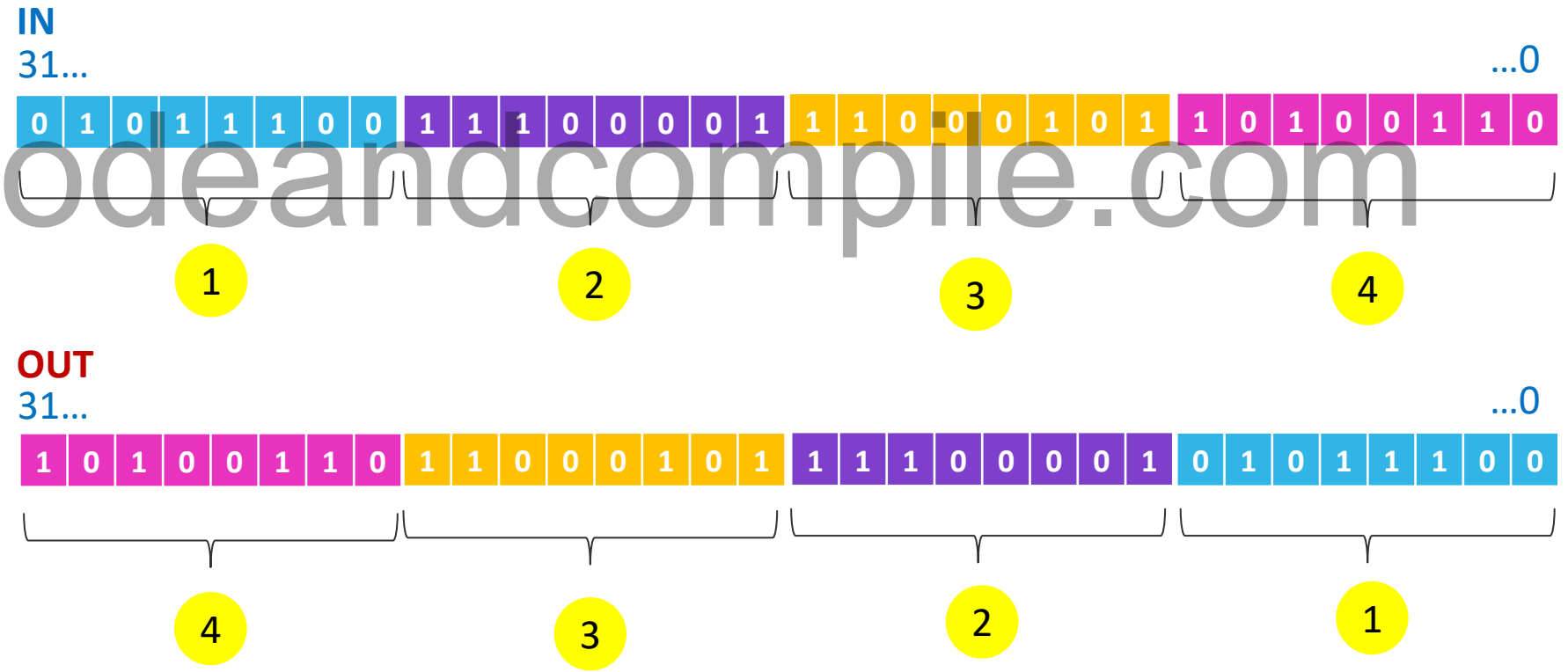


	i	Name	...	Disp...	Monitor value
1		"Block_1".Array_1[0]		DEC	10
2		"Block_1".Array_1[1]		DEC	20
3		"Block_1".Array_1[2]		DEC	45
4		"Block_1".Array_1[3]		DEC	45
5		"Block_1".Array_1[4]		DEC	45
6		"Block_1".Array_1[5]		DEC	45
7		"Block_1".Array_1[6]		DEC	45
8		"Block_1".Array_1[7]		DEC	44
9		"Block_1".Array_1[8]		DEC	99
10		"Block_1".Array_1[9]		DEC	100
11		"Block_1".Array_1[10]		DEC	110

Understanding Move Operations – **SWAP**

SWAP Instruction

You can use the "Swap" instruction to change the order of the bytes at input IN and query the result at output OUT. The following figure shows how the bytes of an operand of the DWORD data type are swapped using the "**Swap**" instruction

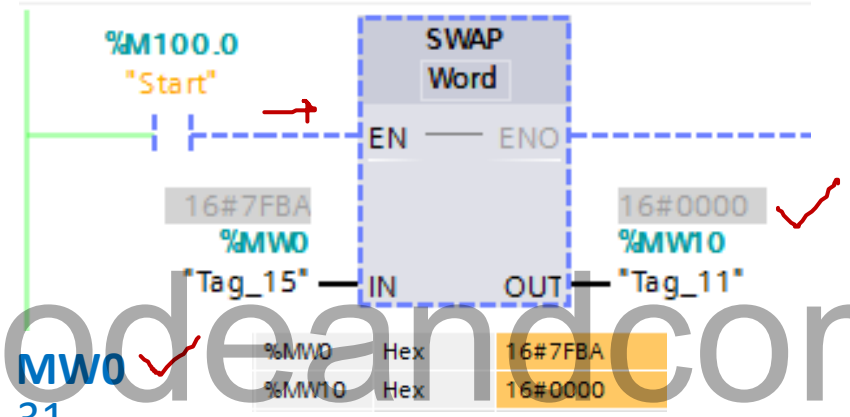


Understanding Move Operations – SWAP

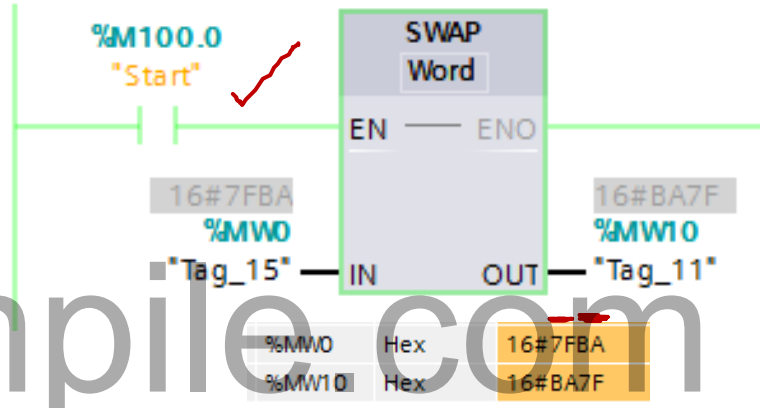
Example:

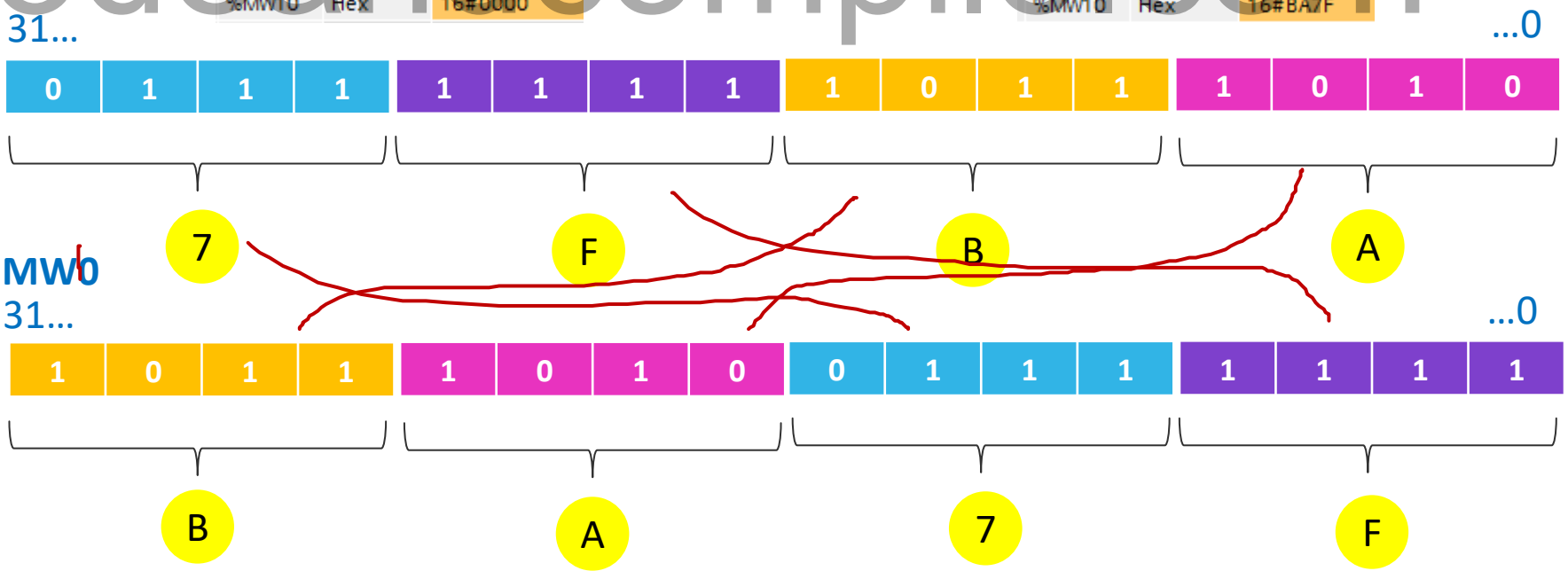
SWAP the values of MW0 and store it in MW10

Before Execution



After Execution

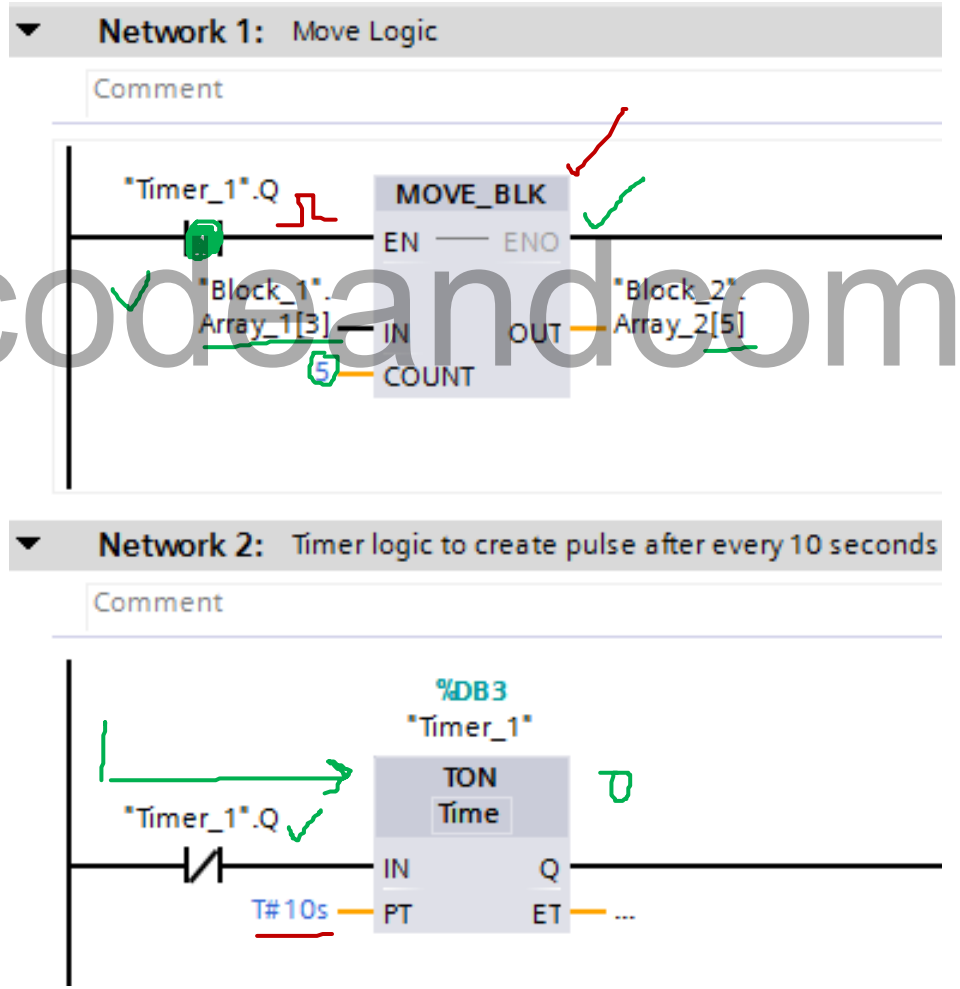




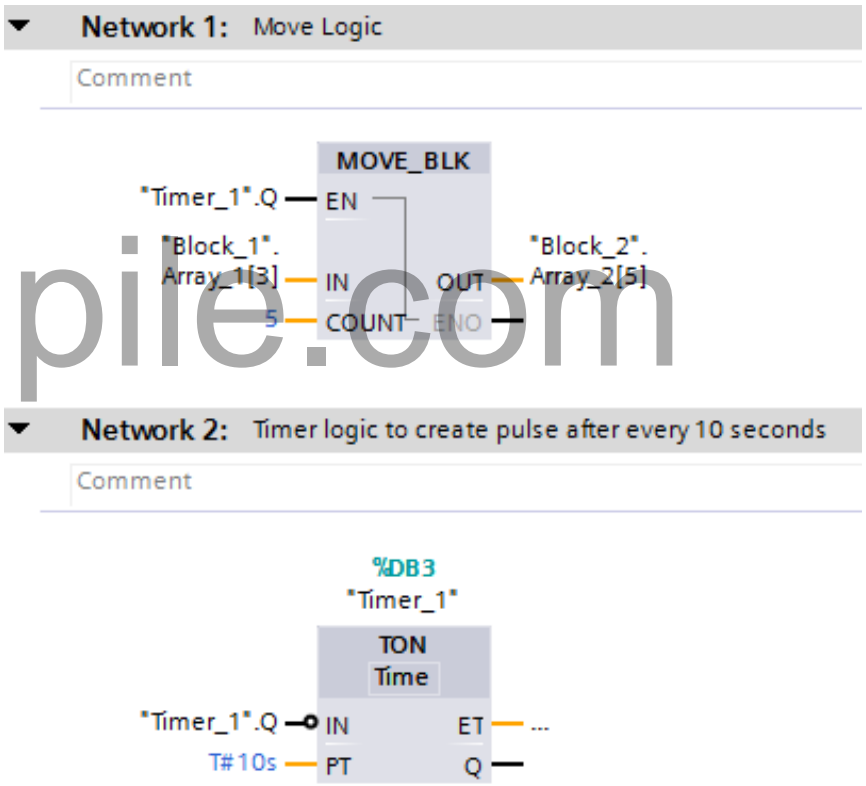
Exercise Example:

Make a **ladder logic** to continuously update the date from **one block** to **another block** after every 10 seconds.

Ladder Logic



FBD Logic



What did we learn today?

- **MOV instruction** is used to copy the data from source to destination
- **MOV_BLK instruction** is used to copy the block of source to block of destination using arrays
- **Fill instruction** is used to fill the destination block with source
- **SWAP instruction** is used to SWAP the data in the data/memory register.

Thank you

Get copy of this presentation in the course!

Code and Compile
Learning Made Easy
www.codeandcompile.com