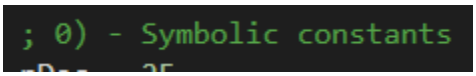
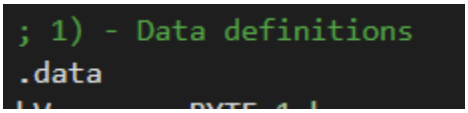
L0010 – Starting Out with Assembly pt 2

Please have your textbook available for reference. (These are based on 3.9.2 and 3.10). Create a new source file and include the following. (Make sure to use comments to show what number the portion of code is answering. See the following images.)





Goal: Practice using Visual Studio and MASM.

0) Define four symbolic constants that represent the integer 25 in decimal, binary, octal, and hexadecimal formats. Place these before the main procedure, after your assembly directives.

Symbolic constants are referred to as **symbols**. They are not stored in program memory, they are used by the assembler to replace values during assembly.

Check your symbols by moving the symbol values into general purpose registers. (i.e. you are moving each of the values you just created into EAX, EBX, ECX, and EDX).

1) Create a definition of each data type listed below. Initialize each variable to a value that is consistent with its data type. Then move each variable into a register and step through your program to make sure that it works. *Note: Make sure that your variable sizes match the register sizes.*

BYTE, SBYTE, WORD, SWORD, DWORD, SDWORD, REAL4

Variables are in memory. To see them, while paused in debug mode, add a Memory window to your view. You can type on addresses to find values. To find a variable by name, use & in front of it. (e.g. for a variable named bVar, use "&bVar").

2) Declare an array of 60 uninitialized unsigned doubleword values. Create another array of 60 unsigned doublewords, initialized to 'abcd'. Look at these arrays in the memory window and note how they are stored. (Intel architecture uses **little-endian** order).

3) Create a DWORD variable in which the hexadecimal data would be stored internally as 12345678.

4) Define a symbol for your name as a null-terminated string.

Tips:

- It may be useful to create or use a template file for new assembly source files, as the directives and procedure setup are always the same.
- Use hexadecimal where possible.
- Try out the watch list in the debugger for checking variables. (It is another window that you can use).