Add Numbers Program 1

The sample programs shown so far in this chapter added integers stored in registers.

Now that you have some understanding of how to declare data, we will revise the same program by making it add the contents of three integer variables and store the sum in a fourth variable.

```
;AddVariables.asm
.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode: DWORD

.data
firstval    DWORD 20002000h
secondval   DWORD 11111111h
thirdval    DWORD 22222222h
sum         DWORD 0

.code
main PROC
    mov eax, firstval
    add eax, secondval
    add eax, thirdval
    mov sum, eax

    INVOKE ExitProcess, 0

main ENDP
END main
```

When you define data using hexadecimal values like in your example, what's stored inside the memory locations are the actual values in the specified format**(DWORD here).** In your case:

DWORD: This specifies that each of these values is a 32-bit (4-byte) integer.

Hexadecimal notation (e.g., 20002000h, 11111111h, 22222222h): These values are written in hexadecimal format, which is a base-16 numbering system. Each digit in hexadecimal represents 4 bits (or half a byte). So, for a DWORD, you have 8 hexadecimal digits, making up 32 bits.

So, when you define firstval, secondval, and thirdval as DWORDs with those hexadecimal values, these exact values are stored in memory as 32-bit integers.

In other words, the values 20002000h, 11111111h, and 22222222h are represented in binary as 32 bits each and are stored accordingly in memory.
