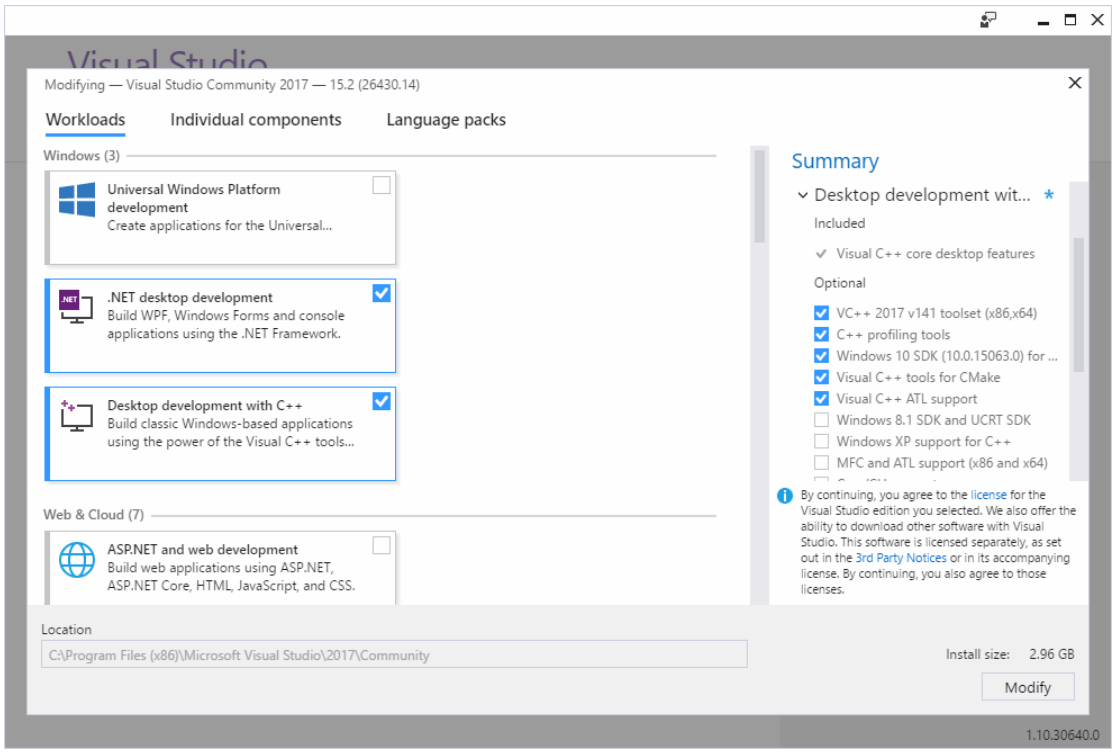
Assembly Language Practice 1

(2024.4)

For assembling 32-bit an assembly program, you need to download visual studio and configure it. Your task is to setup the assembly language environment and write a simple assembly code. Please take the following steps.

1: Download vs2017 or vs2019 or 2022 (https://visualstudio.microsoft.com/downloads，at bottom, select older version)

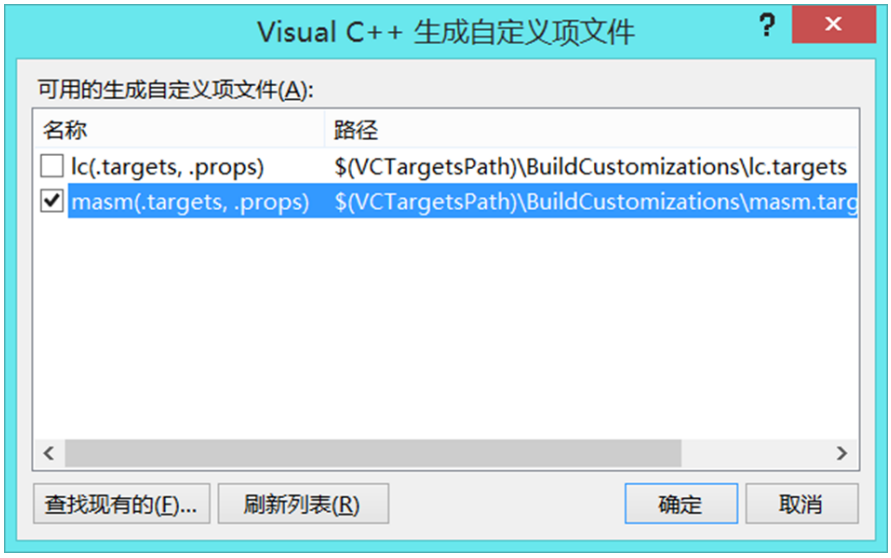
After you have downloaded and installed the VS 2017 or 2019 Community Edition, you may need to install the Visual C++ language and .Net development. If you run the VS installer, select the *Desktop development with C++* button in the installer window, look at the summary list on the right side to verify that VC++ is selected, and click the modify button in the lower right corner of the window.

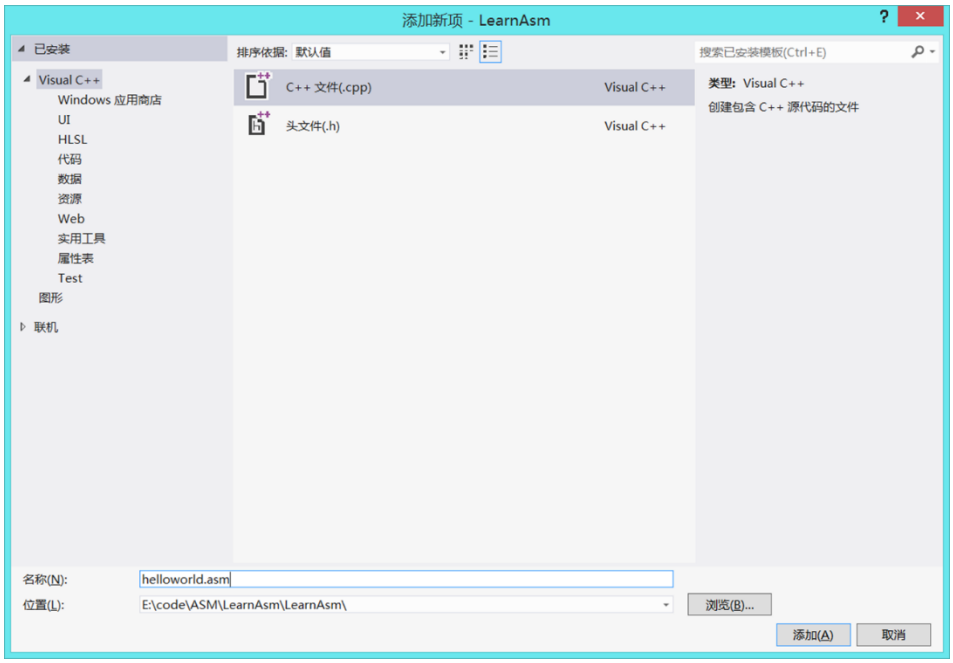


2. Setup Visual studio

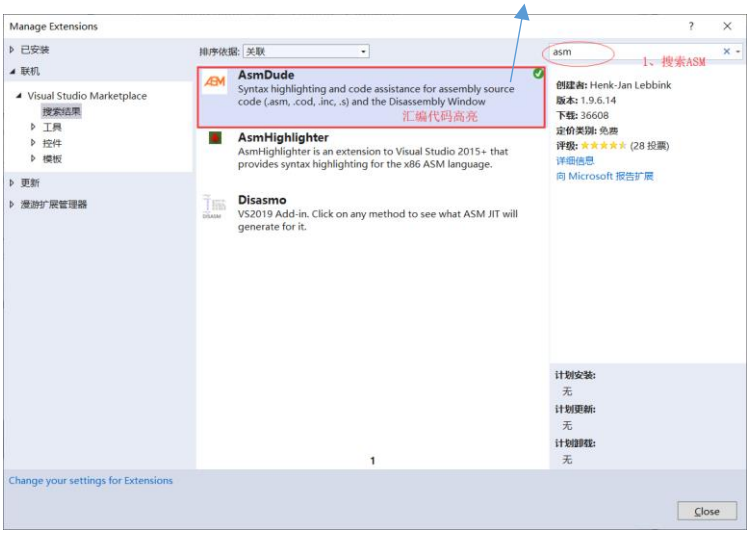
Create a new empty project
Right-click project name to select "create dependent" option and select "masm" option.

Create a new file and select C++ with extension ".asm",



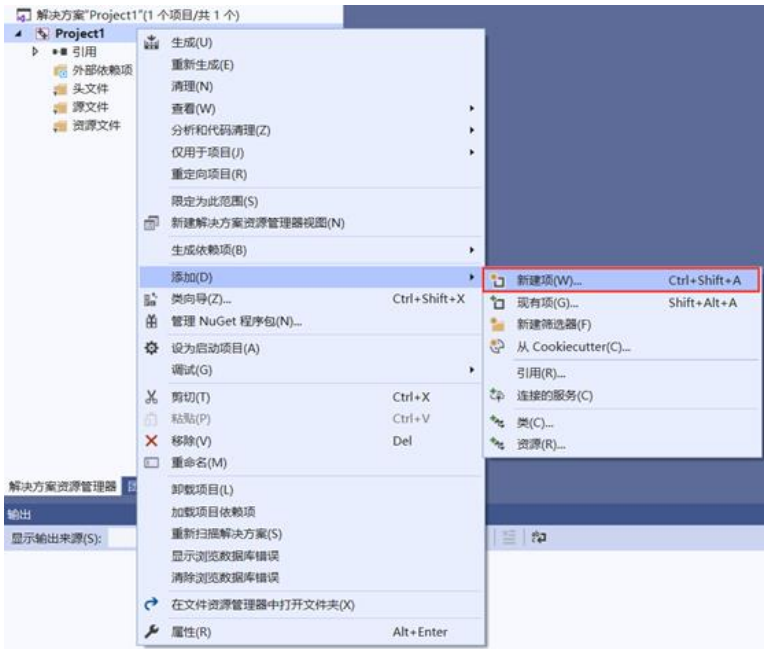


Click tool, Select "expansion and update" option from the tools menu, select "online" option and input "asmdude" in textfield box on the right side and search and install highlight plug-in.



3. Edit, build and run your program

Right-click project name to select "add" option, select "new item".
Input your code in coding workspace.



Build the Program

Now you will build (assemble and link) the sample program. Select **Build Project** from the Build menu. In the Output window for Visual Studio at the bottom of the screen, you should see messages similar to the following, indicating the build

Run your program

Click green arrow(local windows debugger to run your program)

Run the Program in Debug Mode

The easiest way to run your first program is to use the debugger. First, you must set a breakpoint. When you set a breakpoint in a program, you can use the debugger to execute the program a full speed (more or less) until it reaches the breakpoint. At that point, the debugger drops into single-step mode. Here's how to do it:

1. select *Start Debugging* from the Debug menu. The program should run and pause on the line with the breakpoint.

2. Press the F10 key (called *Step Over*) or F11 to execute the current statement. Continue pressing F10 until the program is over.
3. Press F10 one more time to end the program.

4: Write simple assembly program as following

1. Input following asm program and observe the result register and memory in debug mode.

```
; AddTwo.asm - adds two 32-bit integers.
; Chapter 3 example
.386
.model flat,stdcall
.stack 4096
ExitProcess proto,dwExitCode:dword
.code
main proc
 mov eax,5
 add  eax,6
 invoke ExitProcess,0
main endp
end main
```

2: Input the following code and debug to observe the result

```
; AddVariables.asm - Chapter 3 example.
;comment
.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
```

Click debug, select the window, select the register and memory
Select memory, input &sum to view the result

3. Calculate 2^{29}+40 observe the result

4. From the textbook

4.1. Data Definitions

Write a program that contains a definition of each data type listed in Table 3-2 in Section 3.4. Initialize each variable to a value that is consistent with its data type.

4.2 Symbolic Integer Constants

Write a program that defines symbolic constants for all of the days of the week. Create an array variable that uses the symbols as initializers.

4.3 Symbolic Text Constants

Write a program that defines symbolic names for several string literals (characters between quotes). Use each symbolic name in a variable definition.