CSC 3210

Computer Organization and Programming

Lab Work 8

Dr. Zulkar Nine

mnine@gsu.edu

Georgia State University

Summer 2021

Learning Objective

- OFFSET & PTR Operator,
- Checking Memory Data, Type, Length, & Sizeof Operators

Disclaimer

- The process shown in these slides might not work in every single computer due to Operating system version, Microsoft Visual Studio versions and everything.
- If you find any unusual error, you can inform the instructor.
- Instructor will help you resolve the issue.

Attendance!

Lab Work 8 Instructions

- Lab 8(a) : OFFSET operator
- Lab 8(b): PTR operator
- Lab 8(c): Checking memory data
- Lab 8(d): Type, Length and sizeof operator

Due Date: Posted in iCollege

Plan early …

- You have one week time to submit the lab
- Start early
- If you have issues
  - Email TA or instructor
  - Stop by during office hours
- Start working at the last moment is not a good idea.

- Appendix shows how to create a new project.

Directions

- Create a new application for every question.
- Do not use one application with multiple .asm files
- The Appendix has the steps for creating a new project

Submission Instruction

- For each lab (8a to 8d) do the following:
- Debug through each line of code.
  - Execute the instruction
  - Take a screenshot of the code and register window
  - Record the line number, instruction, Register values in the answer sheet.
  - Also add the screenshot
  - **Then explain the register or memory contents.**

Lab 8(a)
OFFSET operator

Submission

OFFSET & PTR Operator

- Write and run a program to find the values of each **destination operand**:

```
.data
    myBytes     BYTE 10h,20h,30h,40h
    myWords     WORD 8Ah,3Bh,72h,44h,66h
    myDoubles DWORD 1,2,3,4,5
    myPointer   DWORD myDoubles

.code
    mov esi, OFFSET myBytes
    mov ax, [esi]                       ; a. AX =2010
    mov eax, DWORD PTR myWords          ; b. EAX =003b008a
    mov esi, myPointer
    mov ax, [esi+2]                     ; c. AX =0000
    mov ax, [esi+6]                     ; d. AX =0000
    mov ax, [esi-4]                     ; e. AX =0044
```

Lab 8(b)
PTR operator

Submission

PTR Operator

- Write and run a program to find the values of each **destination operand**:

```
.data
        varB BYTE  65h,31h,02h,05h
        varW WORD  6543h,1202h
        varD DWORD 12345678h
.code
        mov ax, WORD PTR [varB+2]  ; a. ax=0502
        mov bl, BYTE PTR  varD   ; b. bl = 78
        mov bl, BYTE PTR [varW+2]    ; c. bl =02
        mov ax, WORD PTR [varD+2]  ; d. ax = 1234
        mov eax, DWORD PTR varW   ; e. eax =12026543
```

Lab 8(c)
Checking Memory Data

Submission

Checking Memory Data

- Write and run a program to find the values of a memory location and a register:

```
.data
        dVal DWORD ?
.code
                mov dVal,12345678h
                mov ax,WORD PTR dVal+2
                add ax,3
                mov WORD PTR dVal,ax            ; dVal=
                mov eax,dVal                    ;EAX=
```

See the next slides to find how data in memory are viewed .

Checking Memory Data

- **Problem 3:** Write and run a program to find the values of a memory location and a register:

```
.data
        dVal DWORD ?
.code
                mov dVal,12345678h
                mov ax,WORD PTR dVal+2
                add ax,3
                mov WORD PTR dVal,ax            ; dVal=
                mov eax,dVal                    ;EAX=
```

See the next slides to find how data in memory are viewed .

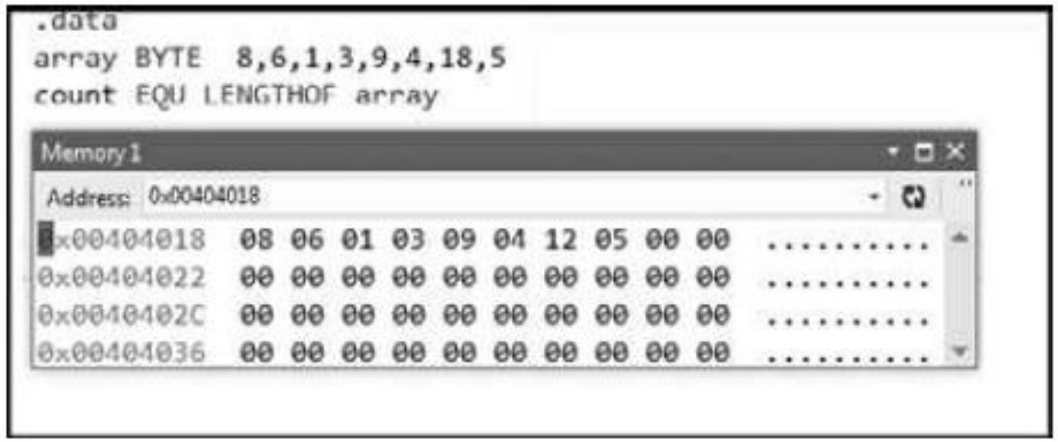
Checking Memory Data

- Use **Memory window** to verify the values of memory locations.
  - **To activate Memory window,** run the debugger, go to debug menu and click on windows, open it, go to Memory then choose **Memory1**.
    - When you run your program and step over every line you will see the changed values marked with red color.

You Must be in the Debugging Mode to see the memory or the register window

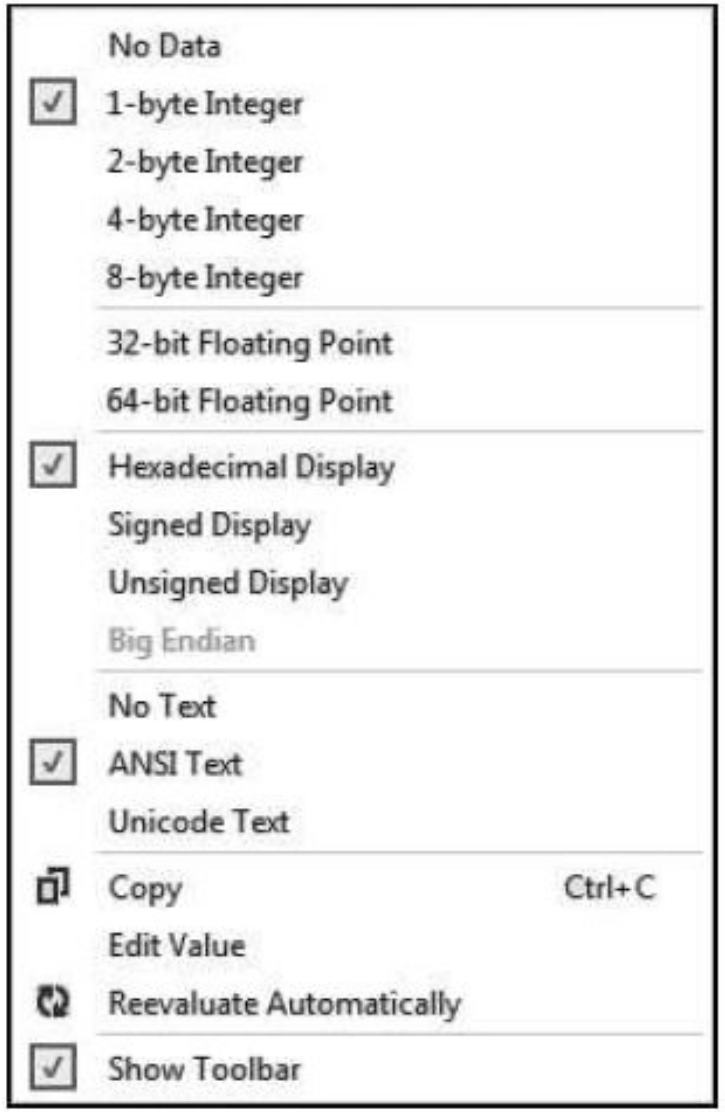
Checking Memory Data

- **To activate Memory window,**
  - if you want to see the location of your variable in the memory,
    - Memory window search box (on the top of the memory window, Address:)
    - write **&** follow it with the variable name: example: **&myVal**l.
    - This will take you to the memory locations of your program (.data section).



Checking Memory Data

- **To activate Memory window,**
  - You can right-click inside the memory window
  - You will see **Popup menu for the debugger's memory window**
  - You can choose how you want to group your bytes: by 1,2,4, or by 8
  - You can also presents data in **hexadecimal**, **signed**, or **unsigned** display



Lab 8(d)
Type, Length, and Sizeof

Submission

Type, Length, & Sizeof Operators

Write and run a program to find the values of each **destination operand**:

```
.data
    myBytes BYTE 10h,20h,30h,40h
    myWords WORD 3 DUP(?),2000h
    myString BYTE "ABCDE"
.code
    mov eax, TYPE myBytes          ; a.
    mov eax, LENGTHOF myBytes          ; b.
    mov eax, SIZEOF myBytes        ; c.
    mov eax, TYPE myWords          ; d.
    mov eax, LENGTHOF myWords          ; e.
    mov eax, SIZEOF myWords        ; f.
    mov eax, SIZEOF myString       ; g.
```

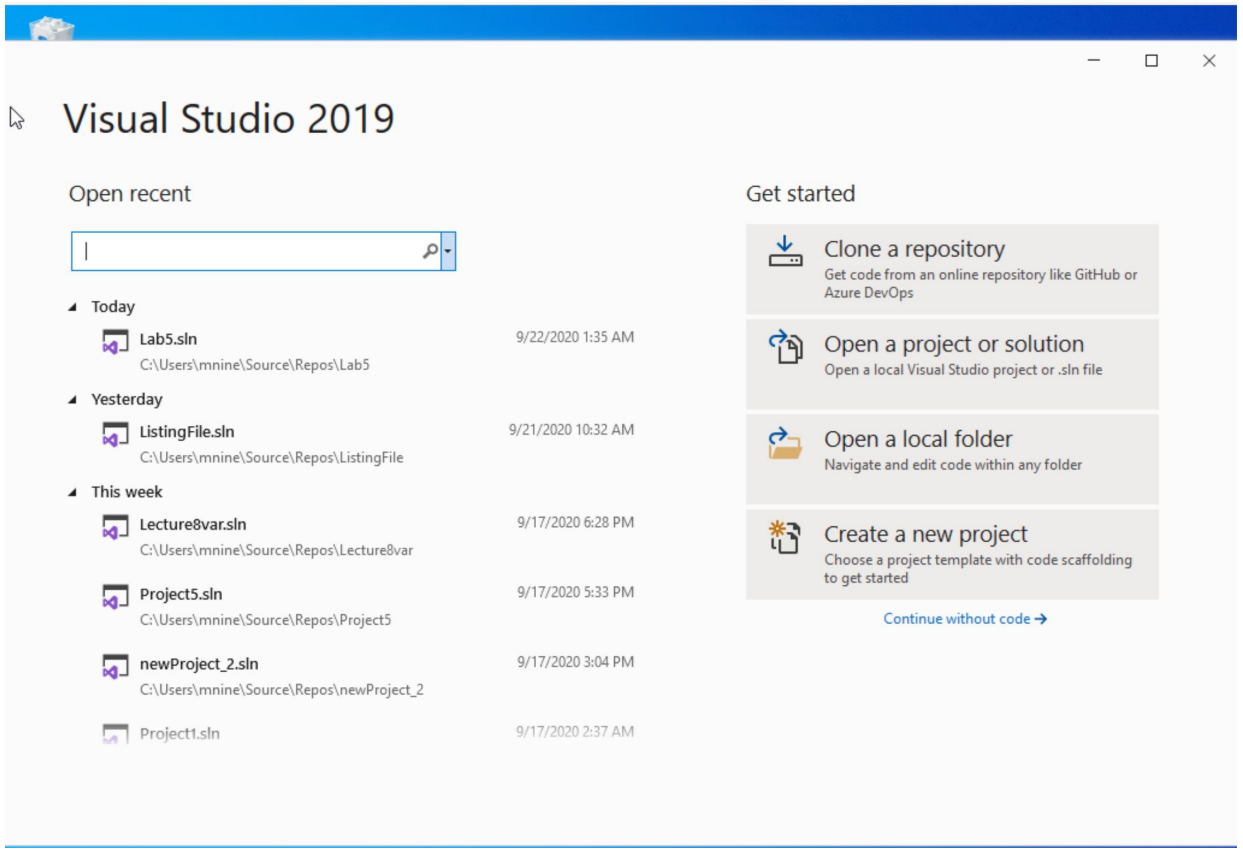
Appendix

Create a Project

Step 1: Create a project (1)

(1) Start Visual Studio

(2) Click Create a new Project

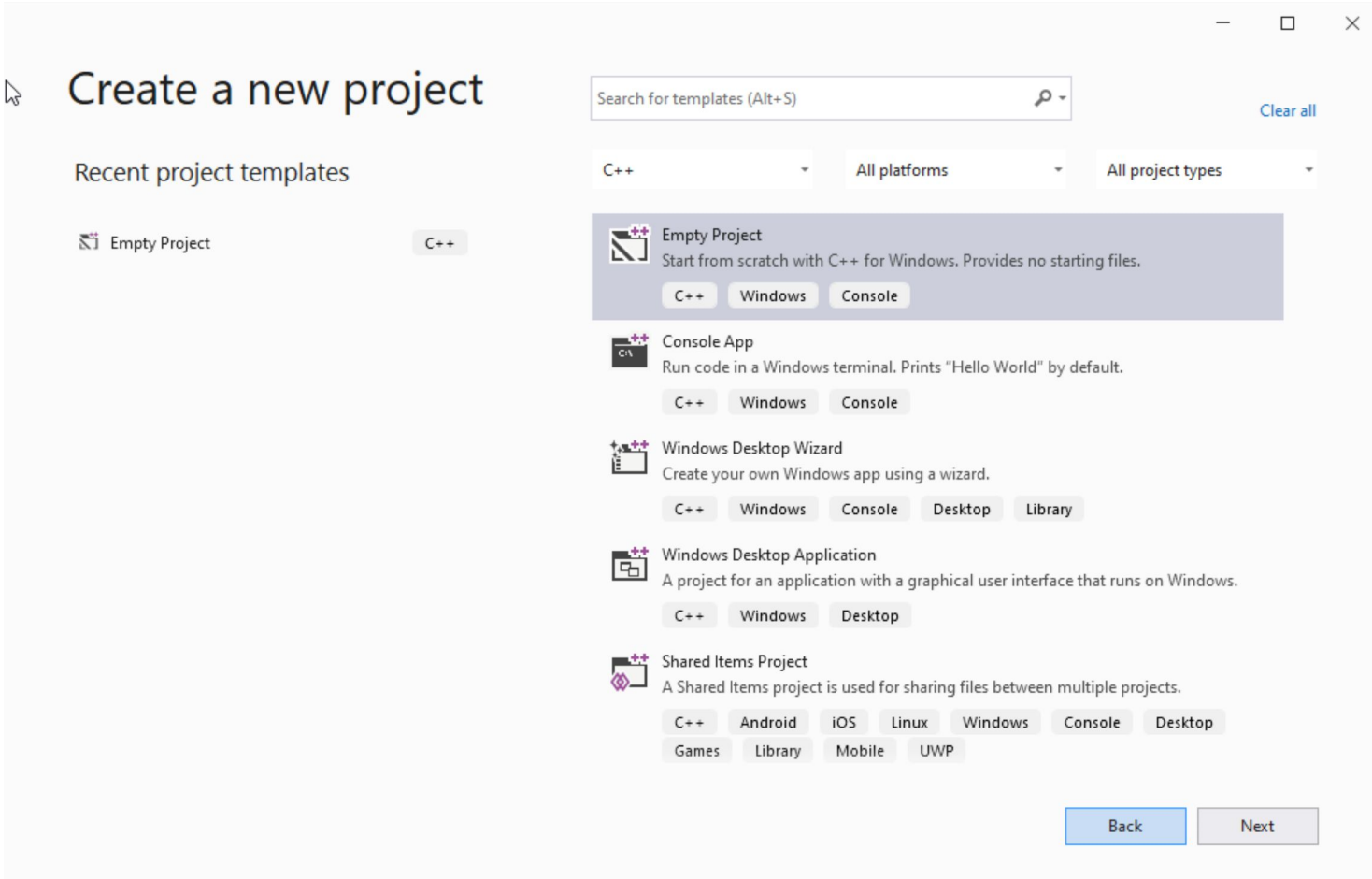


Step 1: Create a project (2)

(1) Select C++ as language

(2) Select Empty Project

(3) Click Next

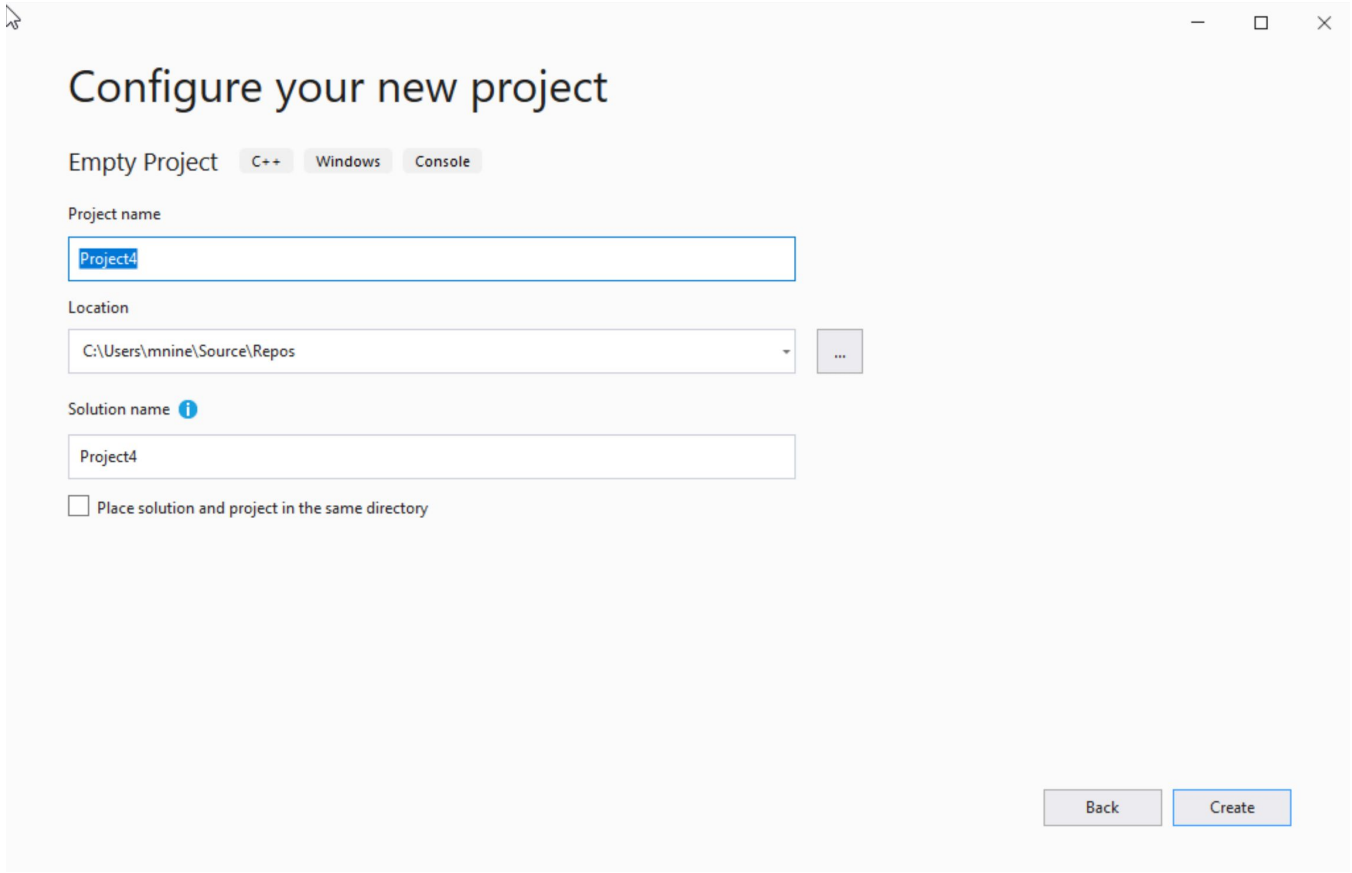


Step 1: Create a project (3)

(1) You can change the project name as you like

(1) Also, you can change the project location

(2) Click Next

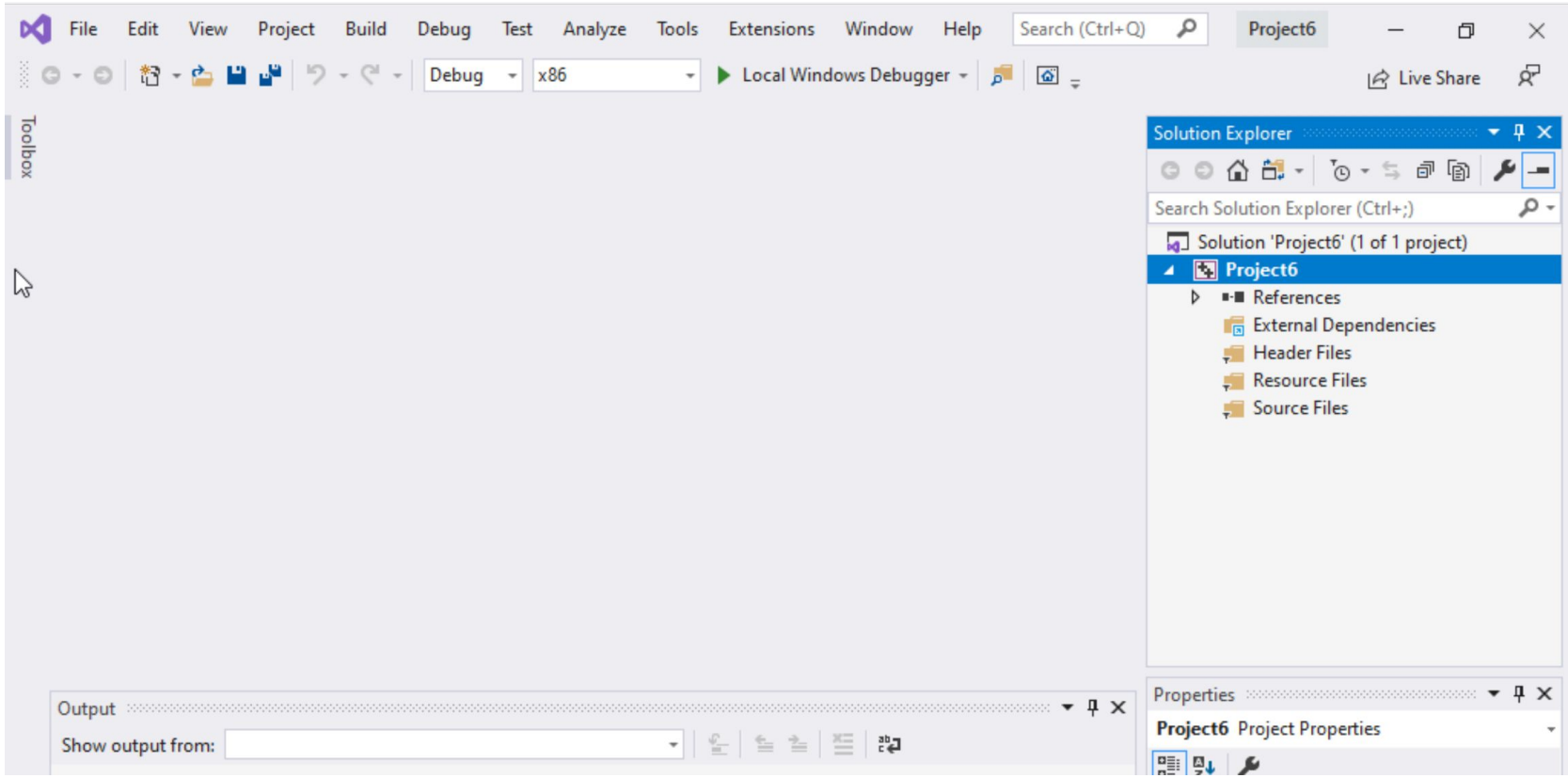


Step 1: Create a project (4)

Delete the

Following folders:

- Header files
- Resources Files, and
- Source Files



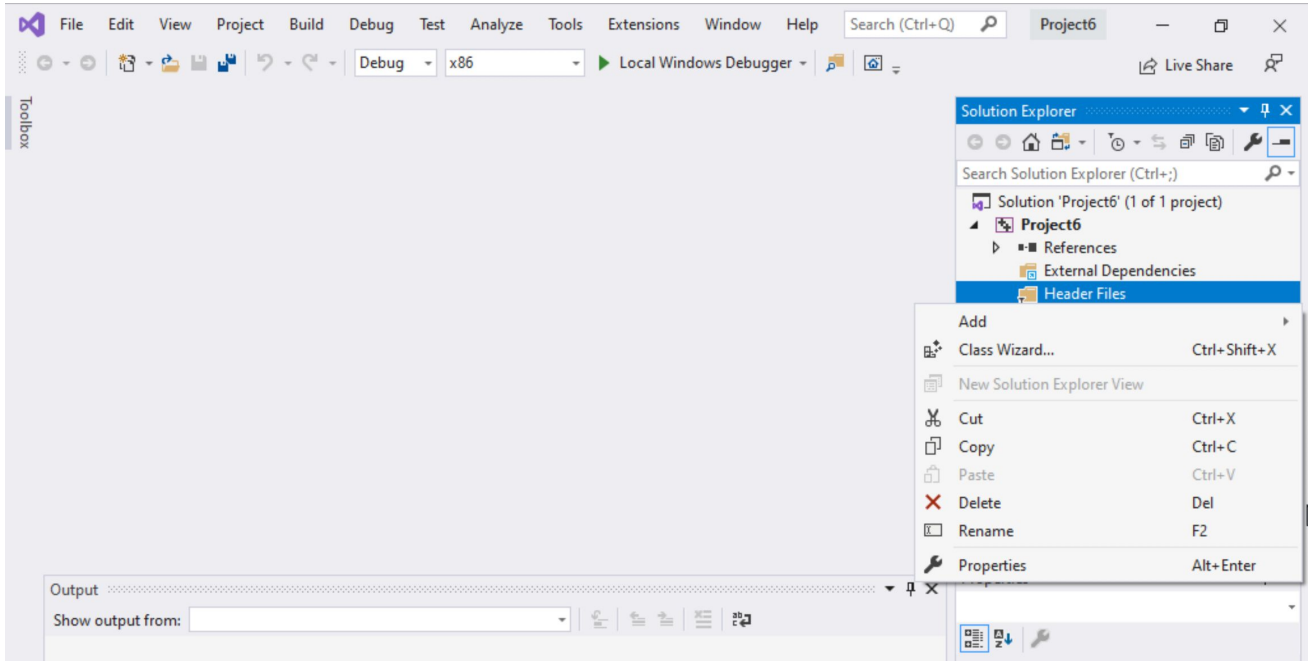
Step 1: Create a project (5)

To delete :

Select the folders

Right click on it

Select delete



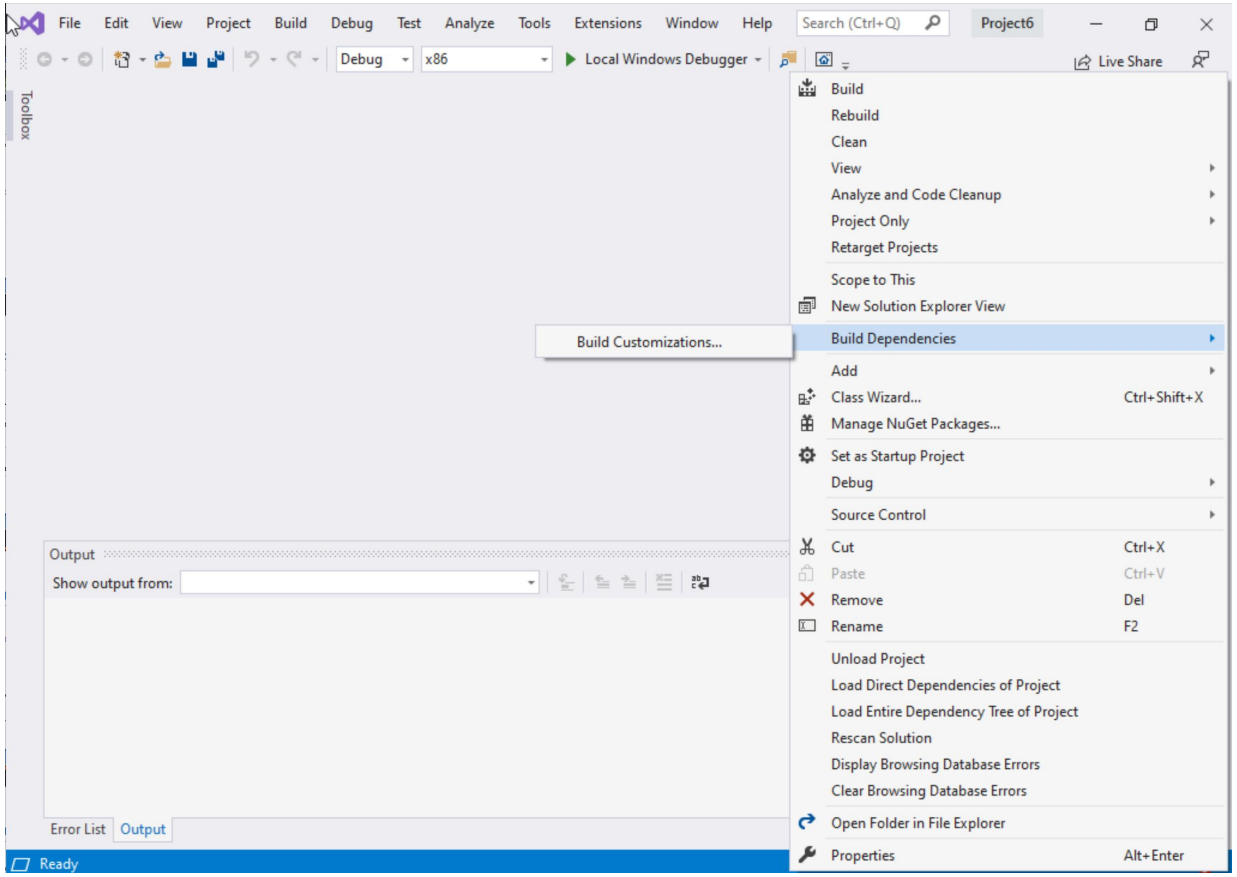
Step 1: Create a project (6)

Select Project Name on solution explorer

Right click on it

Go to Build Dependencies

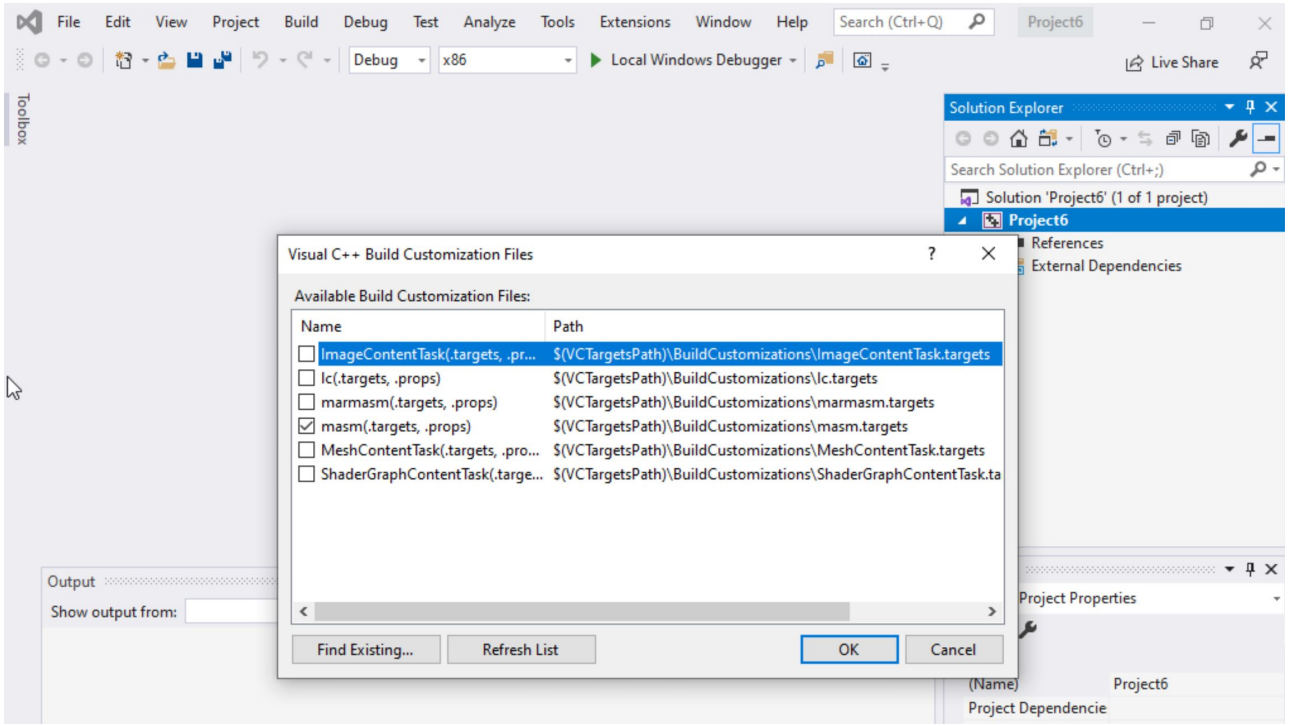
Click on Build Customizations



Step 1: Create a project (7)

Select masm(.target, .props)

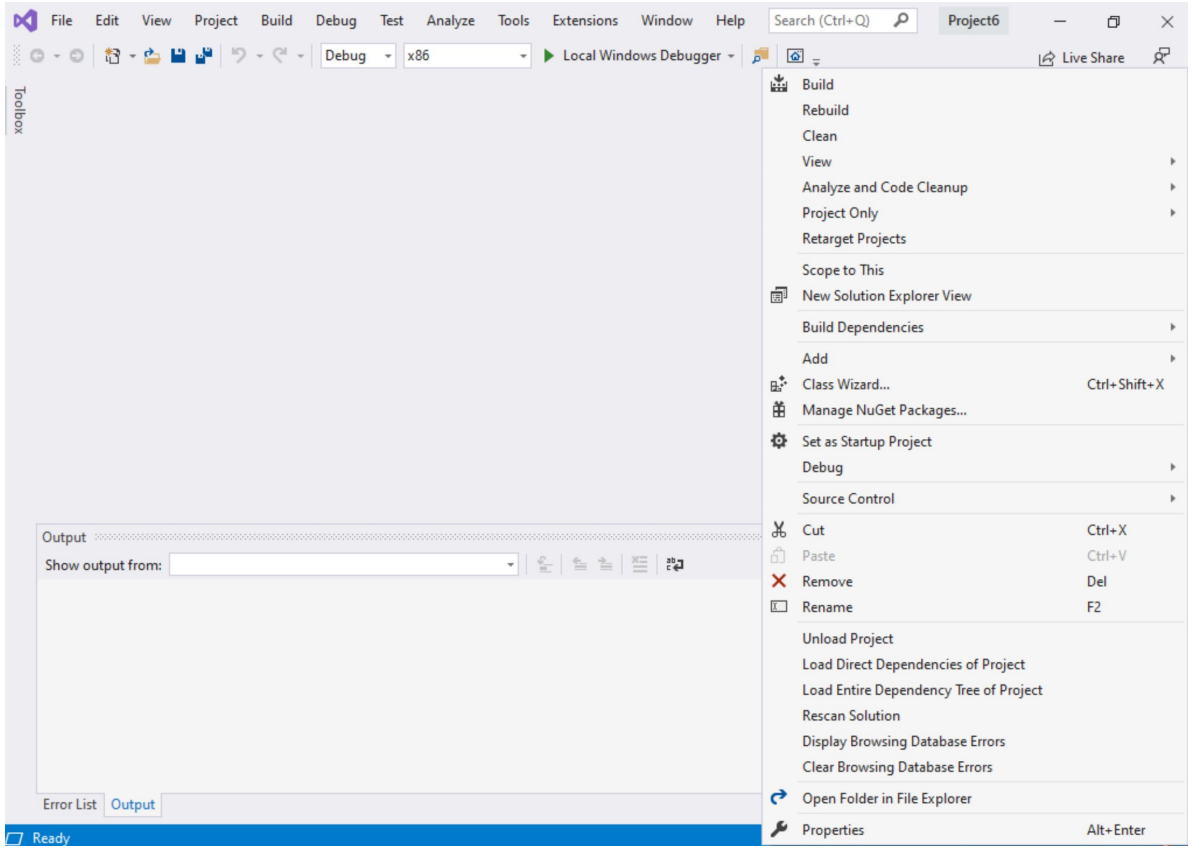
Click ok



Step 1: Create a project (8)

Right click on the Project name in the solution explorer

Click properties



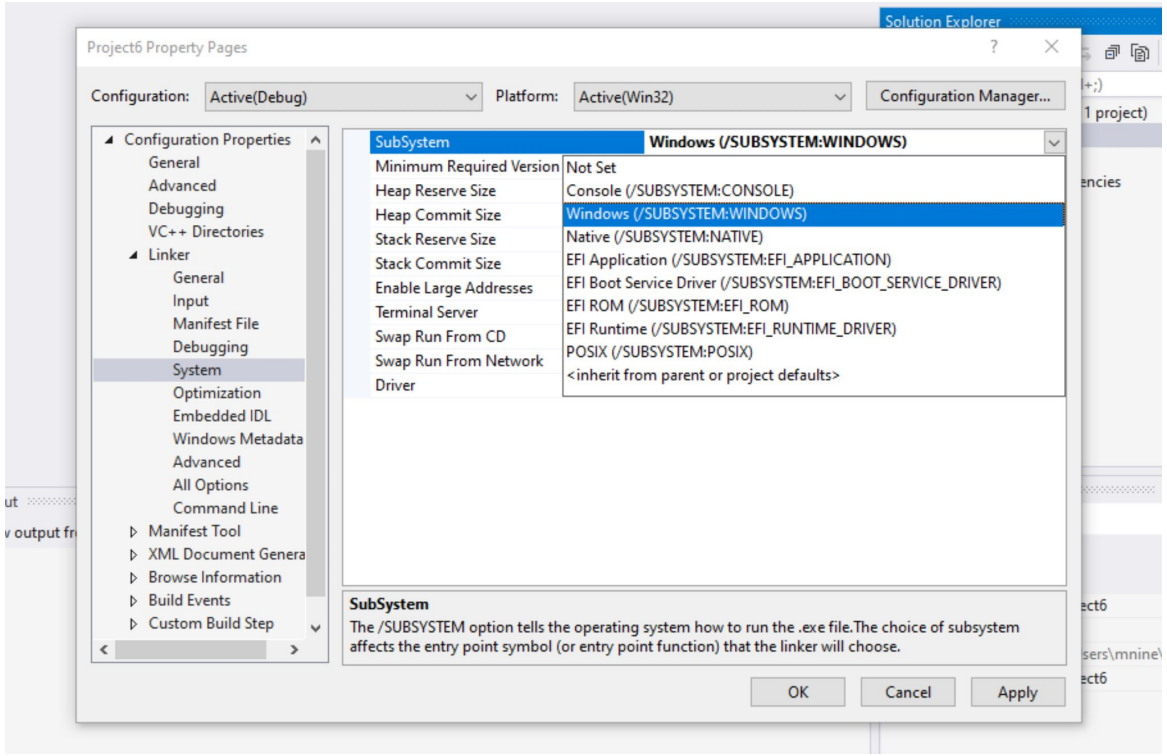
Step 1: Create a project (9)

Expand the ‘Linker’

Select ’System’

Select Windows(/SUBSYSTEM:WINDOWS)

Click OK



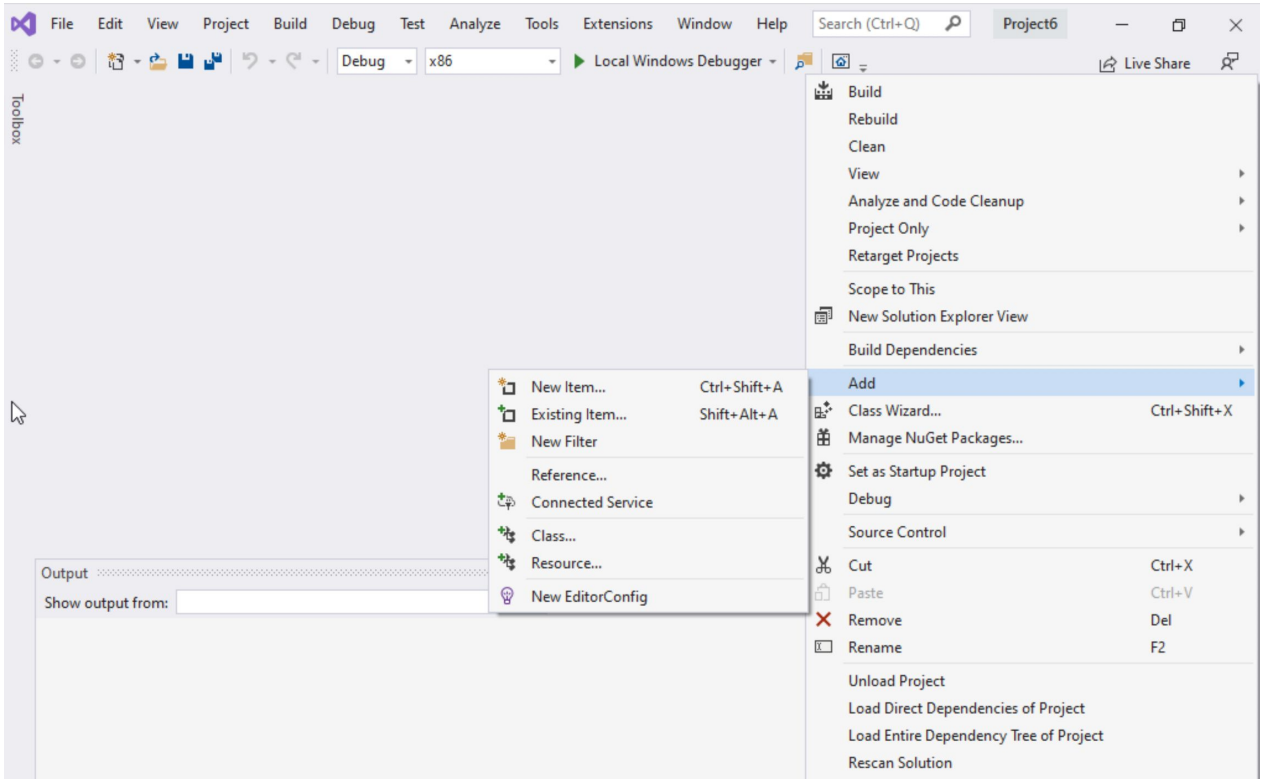
Step 1: Create a project (10)

Select Project name on solution explorer

Right click on it

Expand Add

Choose New Item

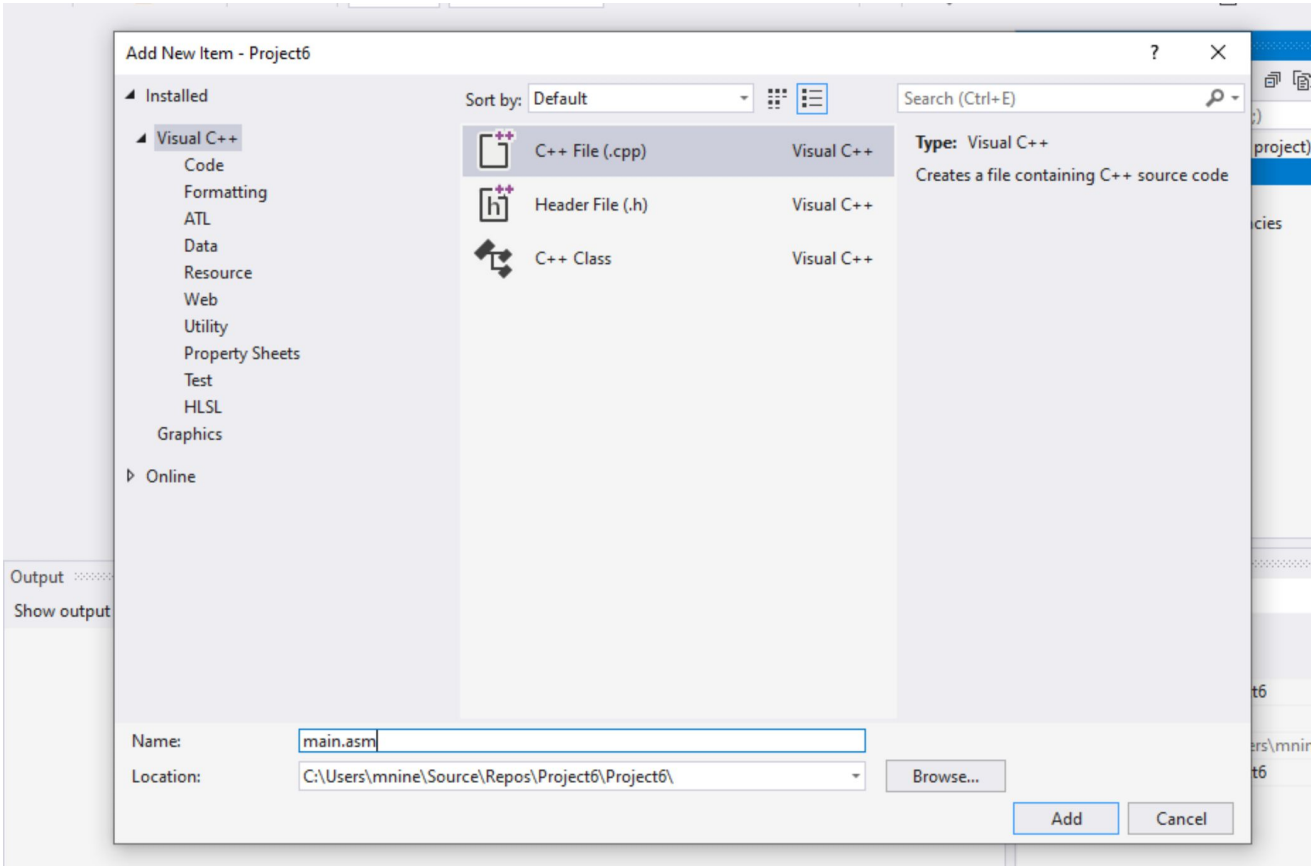


Step 1: Create a project (11)

Select C++ File(.cpp)

Name: main.asm

Click Add



Step 1: Create a project (12)

Select main.asm

Add your code

In the main.asm File.