

CSC 3210

Computer Organization and Programming

Lab Work 7

Dr. Zulkar Nine

mnine@gsu.edu

Georgia State University

Summer 2021

Learning Objective

- Data Transfer,
- Arithmetic instructions with variables

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 7 Instructions

- Lab 7(a) : Mov instructions
- Lab 7(b): Movzx and movsx instructions
- Lab 7(c): Direct-Offset Addressing
- Lab 7(d): xchg instruction
- Lab 7(e): Expression evaluation

Due Date: Posted on 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.

Lab 7(a)

Mov instruction

Submission

Data Transfer Instruction: Mov

- Create a new Project to run the following program.
- Examine the error message generated by MASM assembler
- Do the visual studio errors' messages show the type of error?
- Correct those errors
- Build and run the program using the debugger

```
.data  
count BYTE 100  
wVal WORD 2  
wVal2 word 4  
.code  
    mov bl, count  
    mov ax, wVal  
    mov count,al  
  
    mov al, wVal  
    mov ax, count  
    mov eax, count  
    mov wVal2,wVal
```


Submission Instruction 7(a)

- There is an answer sheet attached to the lab
 - After fixing errors, build your project without errors.
 - Take a screenshot of code and the message showing that build is successful.

Lab 7(b)

Submission

Movzx and Movsx

- Create a new project to run the following program.
- Build and run the program using the debugger
- Examine the content of the registers

```
.data  
myByte1 BYTE 9Bh  
.code  
    mov bx, 0A69Bh  
    movzx eax, bx      ;EAX =  
  
    movzx eax, myByte1 ;EAX=  
  
    mov bx,0A69Bh  
    movsx eax,bx      ;EAX =
```

Submission Instruction

- Debug through each line of your 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 contents.

Lab 7(c)

Submission

Data Transfer Instructions: Direct-Offset Addressing

- Create a new Project to run the following program.
- Build and run the program using the debugger
- Examine the content of the registers

.data

arrayD DWORD 10000h,20000h, 30000h

.code

; Direct-Offset Addressing (doubleword array):

mov eax,arrayD ; EAX =

mov ebx,[arrayD+4] ; EBX =

mov edx,[arrayD+8] ; EDX =

Submission Instruction

- Debug through each line of your 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 contents.

Xchg instruction

A review

XCHG instruction

.data

val1 WORD 1000h

val2 WORD 2000h

.code

; Memory-to-memory exchange:

mov ax,val1 ; AX =

xchg ax,val2 ; AX= , val2=

mov val1,ax ; val1 =

Exchange item in an array

- Create a new project to run the following program.
- Build and run the program using the debugger
- Examine the content of the registers
 - rearranges the values of three **doubleword** values in the following array as:

1,2,3 -----□ 3, 1, 2.

See next slide - □

Data Transfer Instructions : Direct-Offset Operands

.data

arrayD **DWORD** 1,2,3

- **Step1:** copy the **FIRST** value into **EAX** and **exchange** it with the value in **the SECOND position**.

```
mov eax, arrayD  
xchg eax, [arrayD+4]
```

```
XCHG  reg, reg  
XCHG  reg, mem  
XCHG  mem, reg
```

- **Step 2:** Exchange **EAX** with the **THIRD** array value and copy the value in **EAX** to the **FIRST** array position.

```
xchg eax, [arrayD+8]  
mov arrayD, eax
```

Lab 7(d)

Submission

Submission instruction.

- Create a new project to run the following program
- Declare an array:
 - `arrayB WORD 1,2,3,4`
- Write code to Rearrange the array as follows:
 - 4,3,1,2
- Debug the code until you reach “INVOKE ExitProcess, 0”
- Add the screenshot of your code in the answer sheet.

Lab 7(e)

Submission

Submission Instruction

- Create a new application to run the following program.
- The data segment is provided:

.data

Val1 SWORD 23

Val2 SWORD -35

Val3 SDWORD 4

- Evaluate the following expression :

$EBX = (-Val1 + val2) + (val3 * 3)$

You can only use Mov, Movsz, Movzx, Add, Sub instructions.

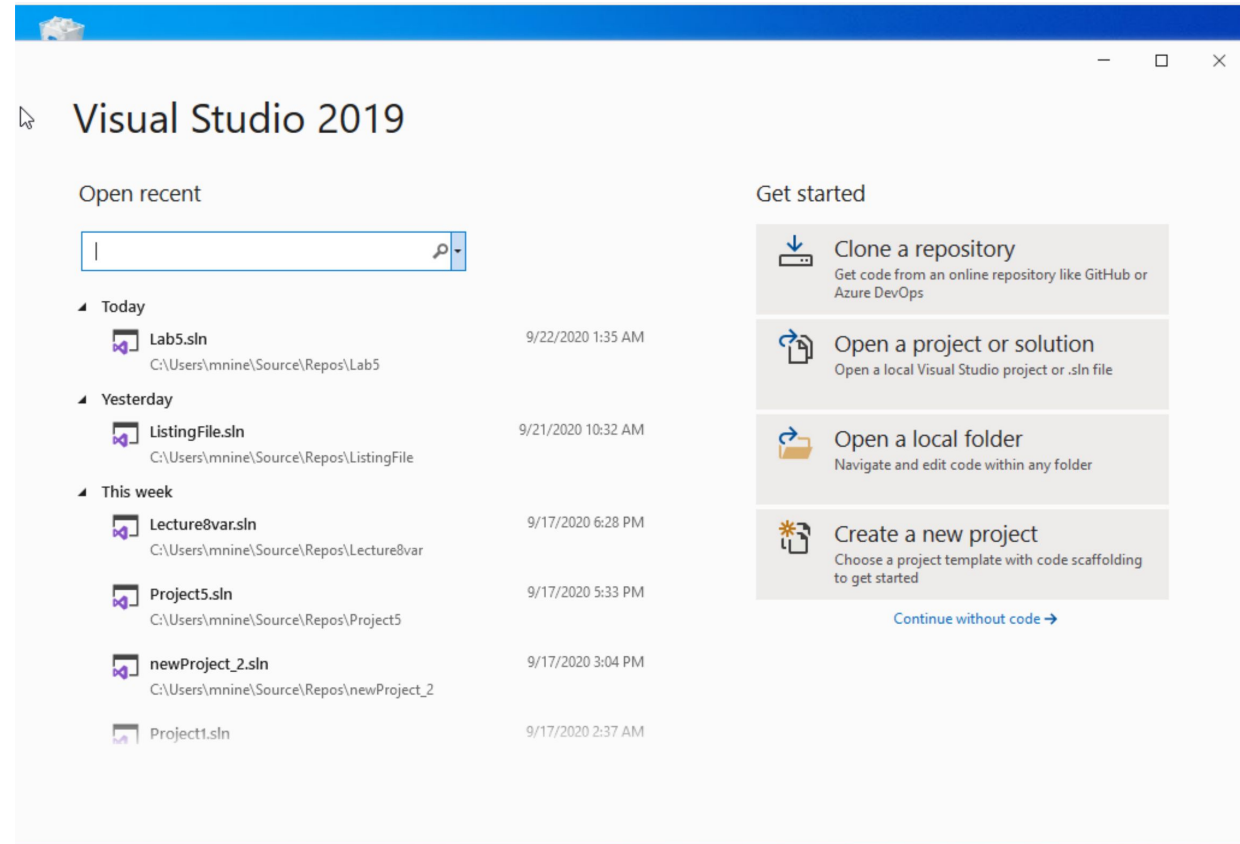
- Build and run the program using the debugger
- Examine the content of the registers

Appendix

Create a Project

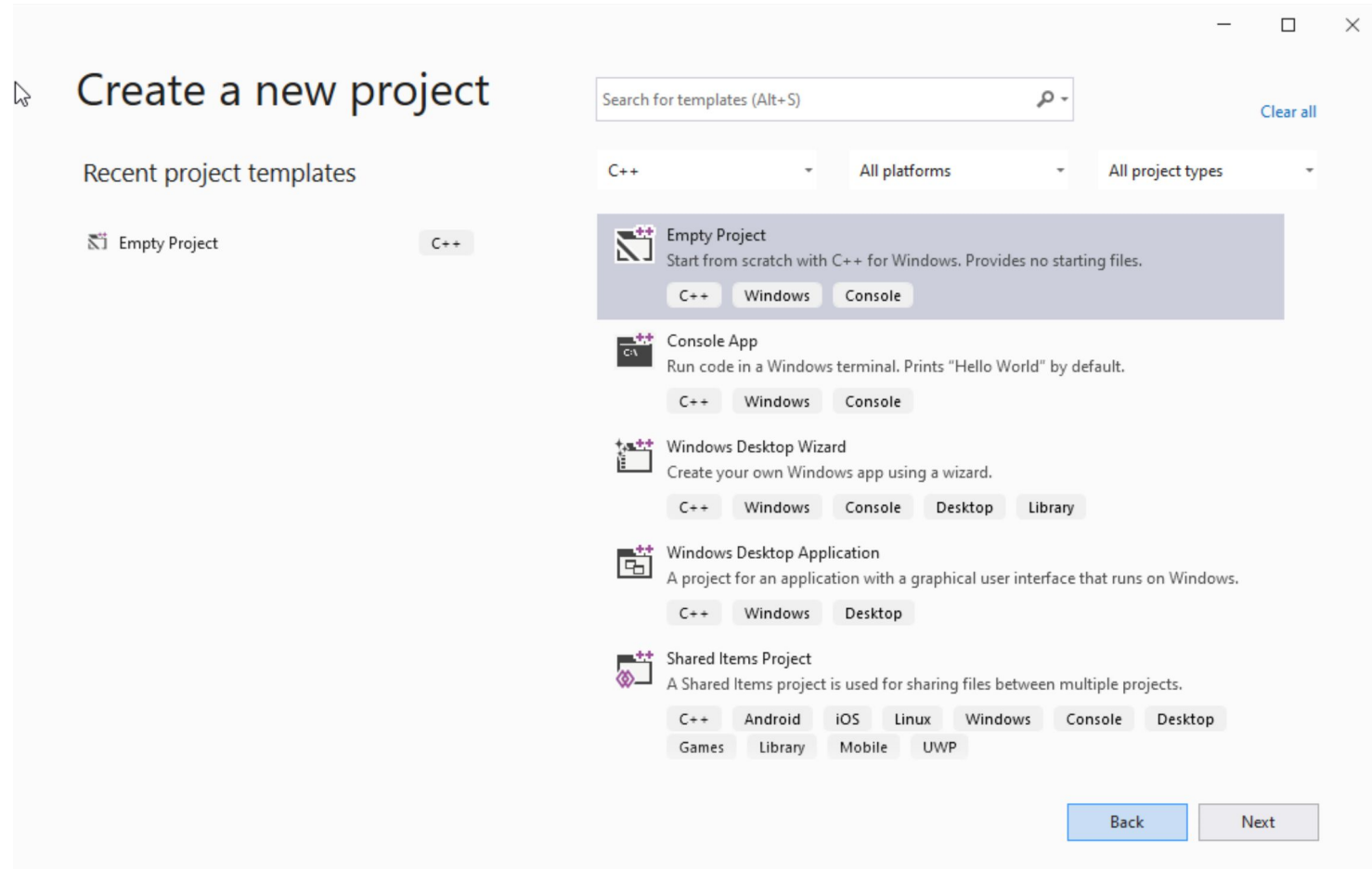
Step 1: Create a project (1)

- (1) Start Visual Studio
- (2) Click Create a new Project



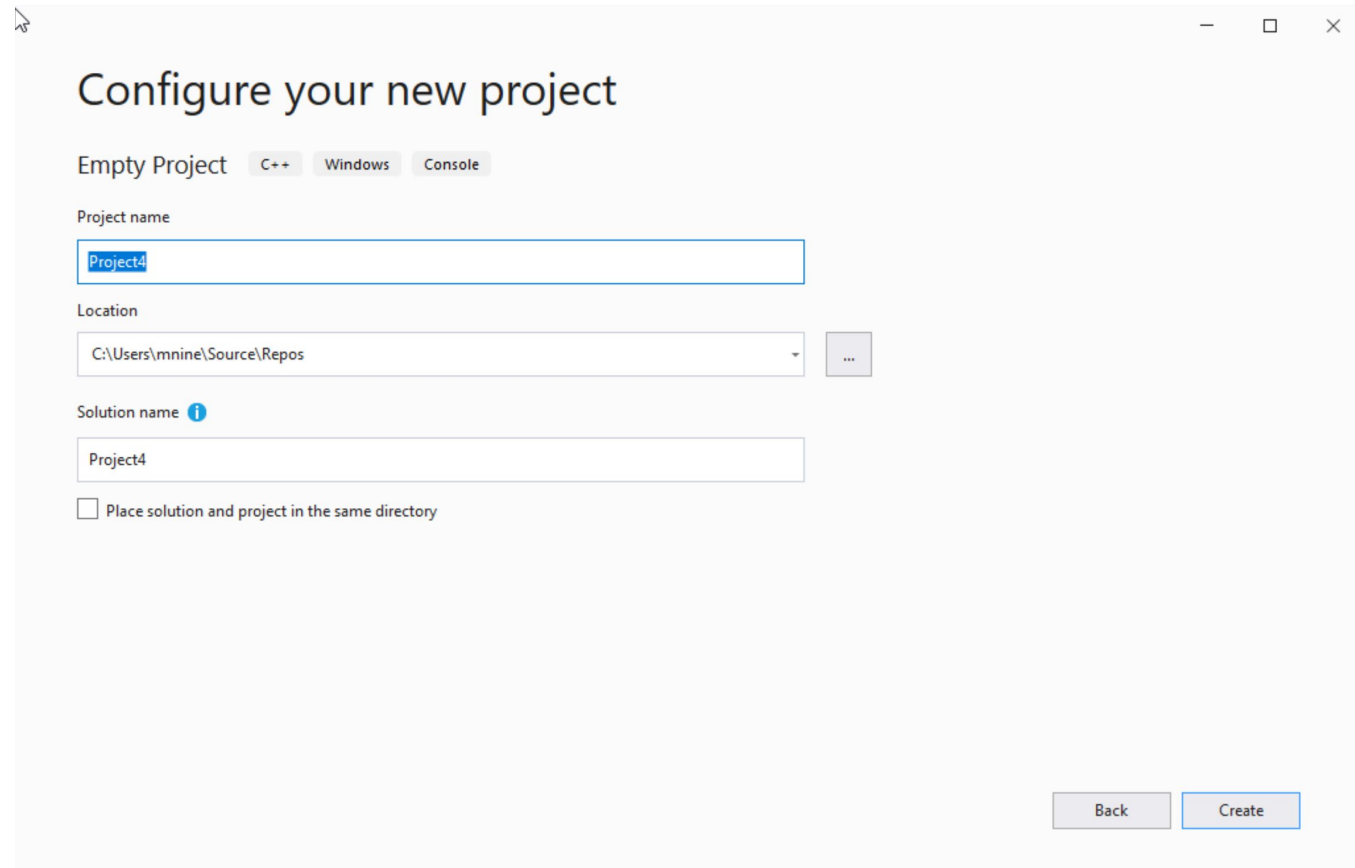
Step 1: Create a project (2)

- (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



The screenshot shows the 'Configure your new project' dialog box in Visual Studio. The title bar includes standard window controls. The main heading is 'Configure your new project'. Below it, there are tabs for 'Empty Project', 'C++', 'Windows', and 'Console'. The 'Empty Project' tab is selected. The 'Project name' field contains 'Project4'. The 'Location' field shows the path 'C:\Users\mnine\Source\Repos' with a browse button ('...') to its right. The 'Solution name' field, which has an information icon, also contains 'Project4'. At the bottom, there is a checkbox labeled 'Place solution and project in the same directory' which is currently unchecked. In the bottom right corner, there are 'Back' and 'Create' buttons.

Step 1: Create a project (4)

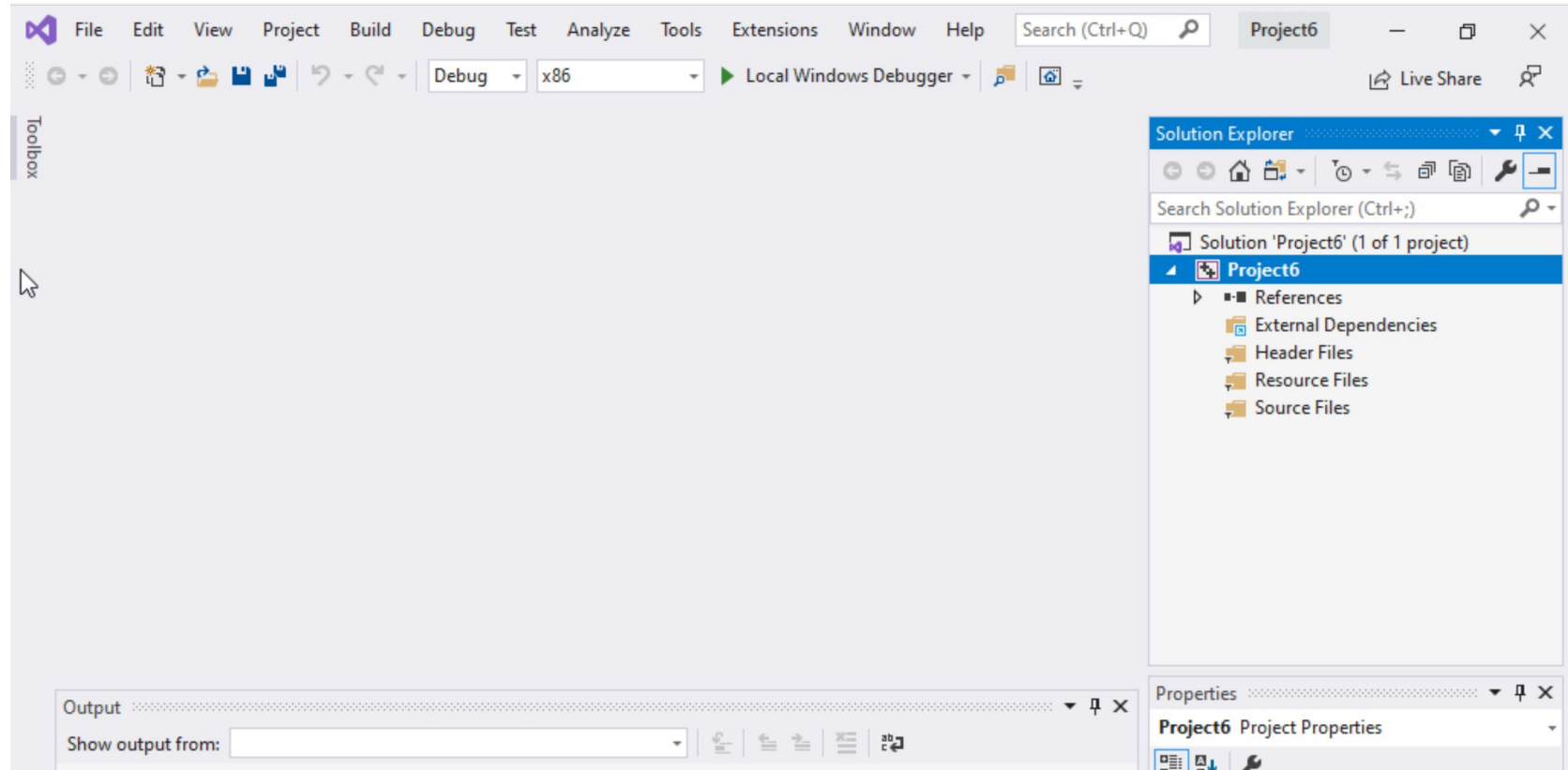
Delete the

Following folders:

- Header files

- Resources Files, and

- Source Files



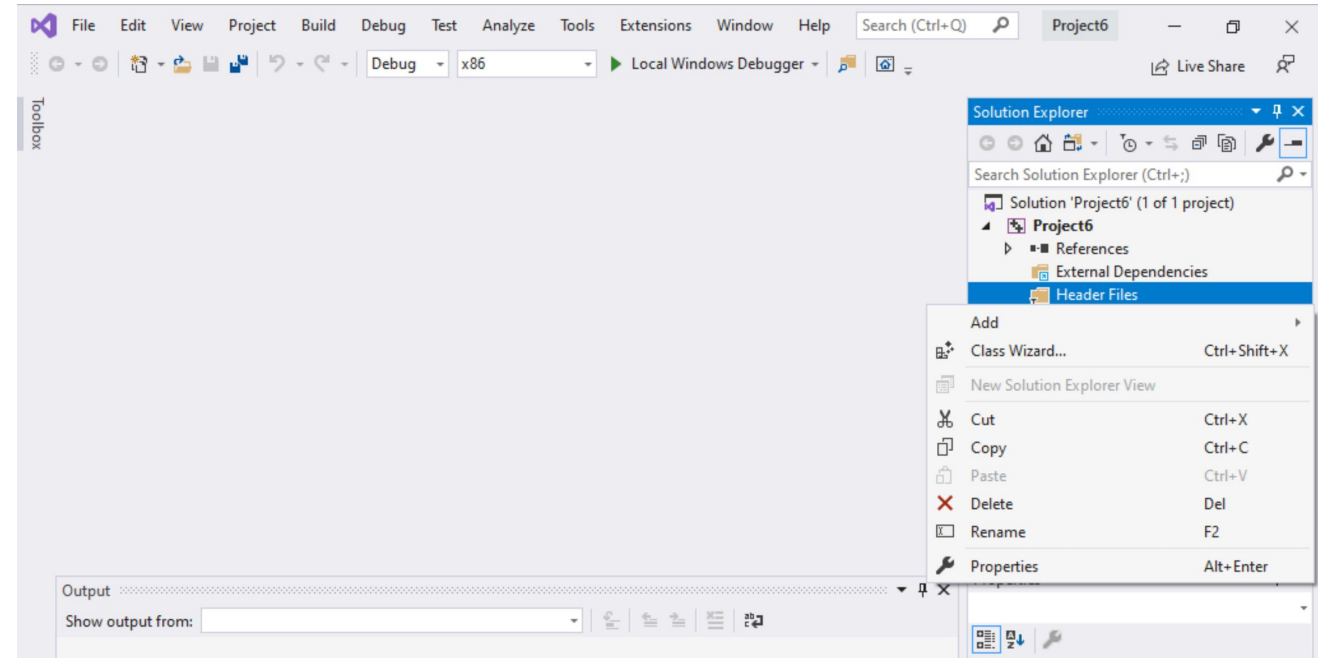
Step 1: Create a project (5)

To delete :

Select the folders

Right click on it

Select delete



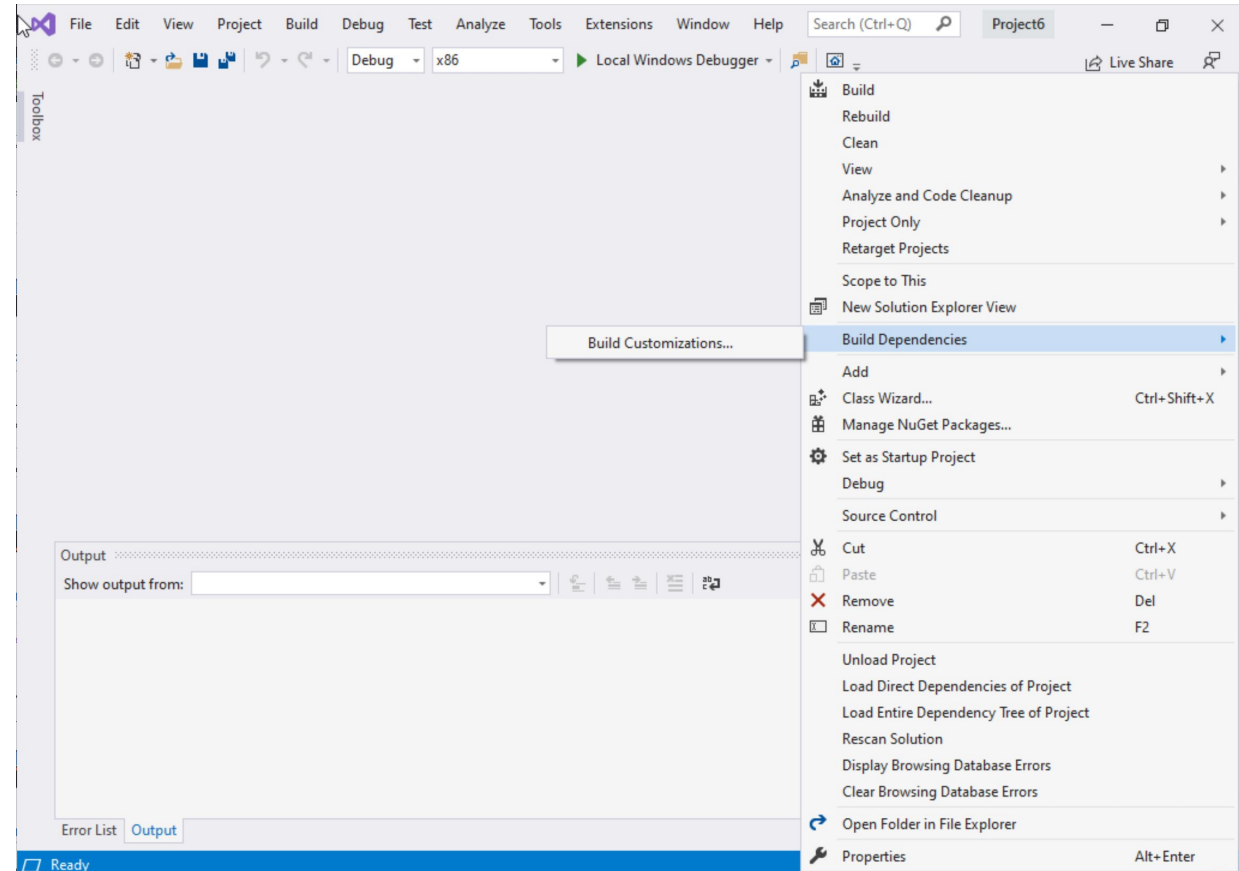
Step 1: Create a project (6)

Select Project Name on solution explorer

Right click on it

Go to Build Dependencies

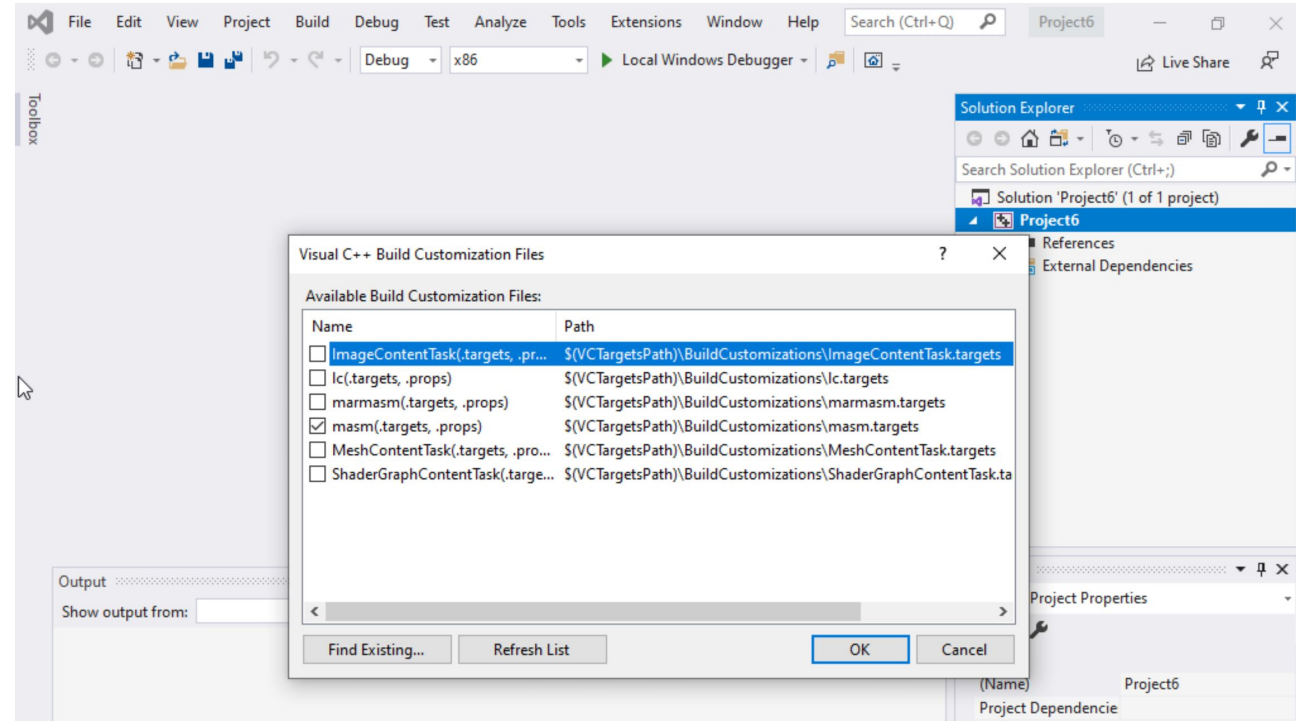
Click on Build Customizations



Step 1: Create a project (7)

Select masm(.target, .props)

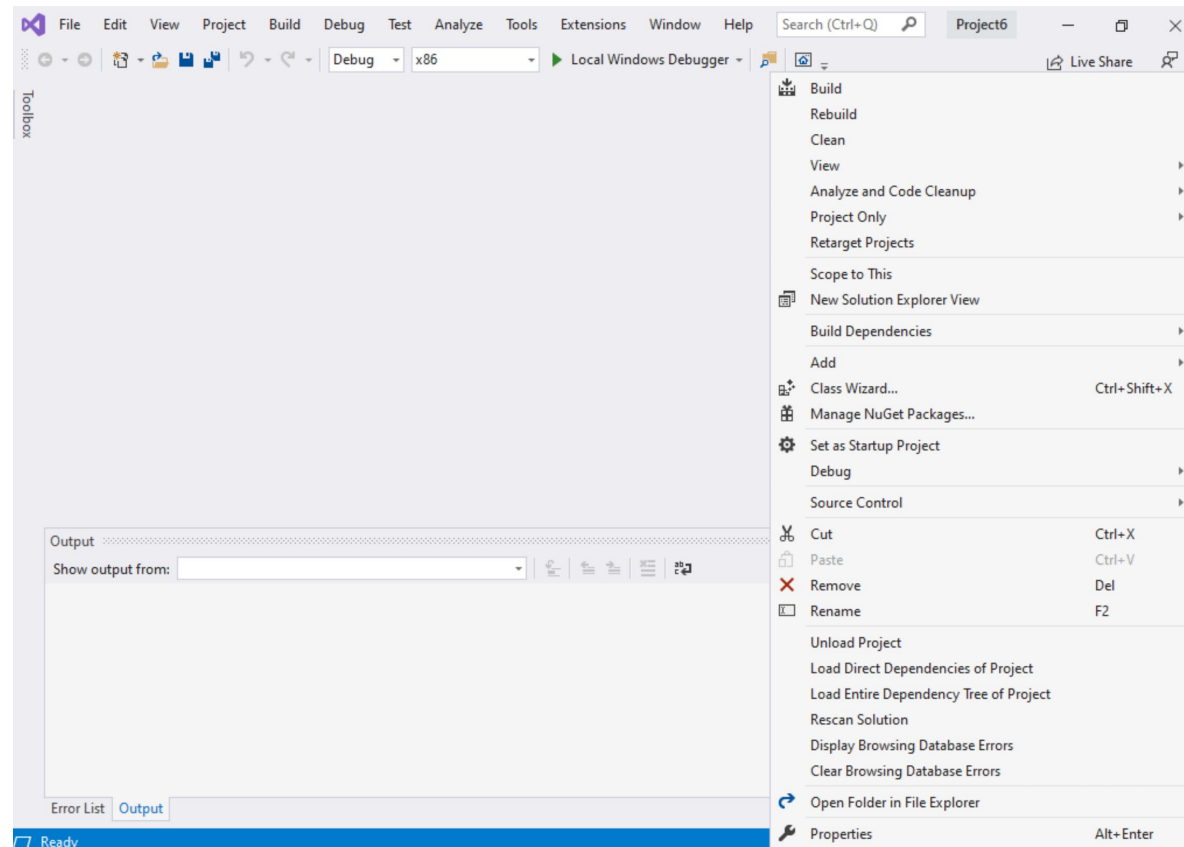
Click ok



Step 1: Create a project (8)

Right click on the Project name in the solution explorer

Click properties



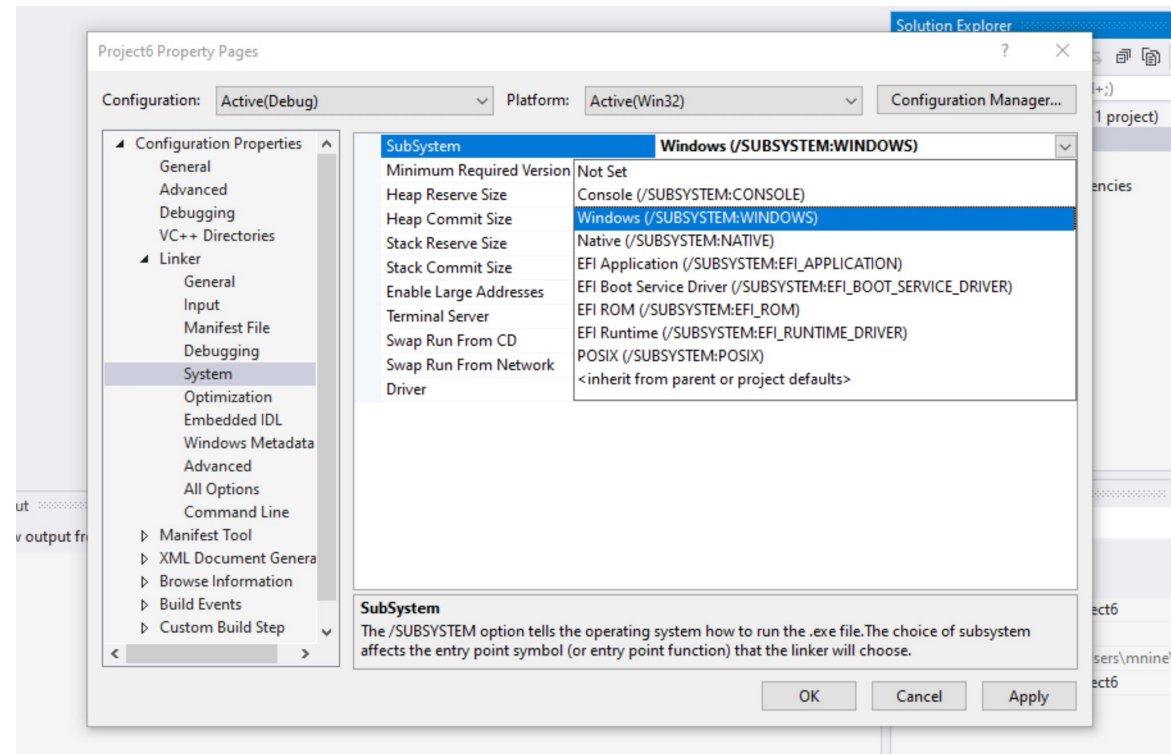
Step 1: Create a project (9)

Expand the 'Linker'

Select 'System'

Select Windows(/SUBSYSTEM:WINDOWS)

Click OK



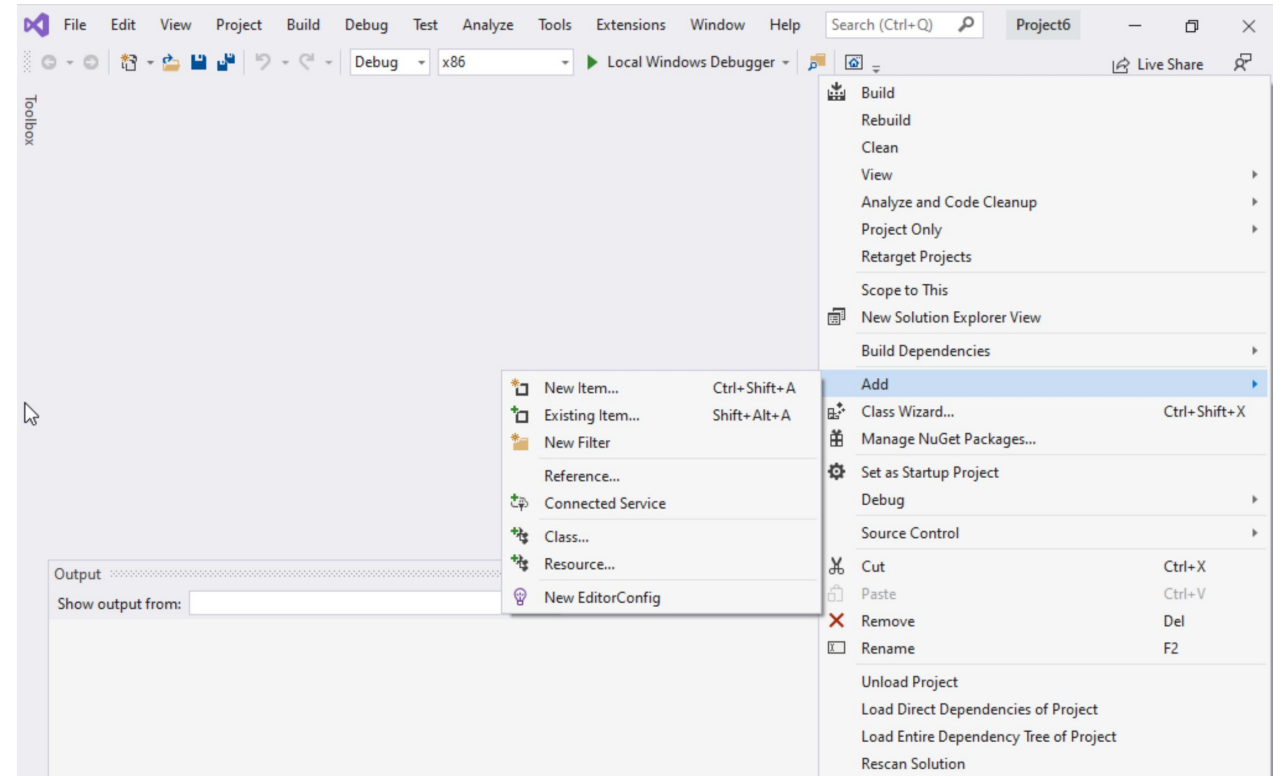
Step 1: Create a project (10)

Select Project name on solution explorer

Right click on it

Expand Add

Choose New Item

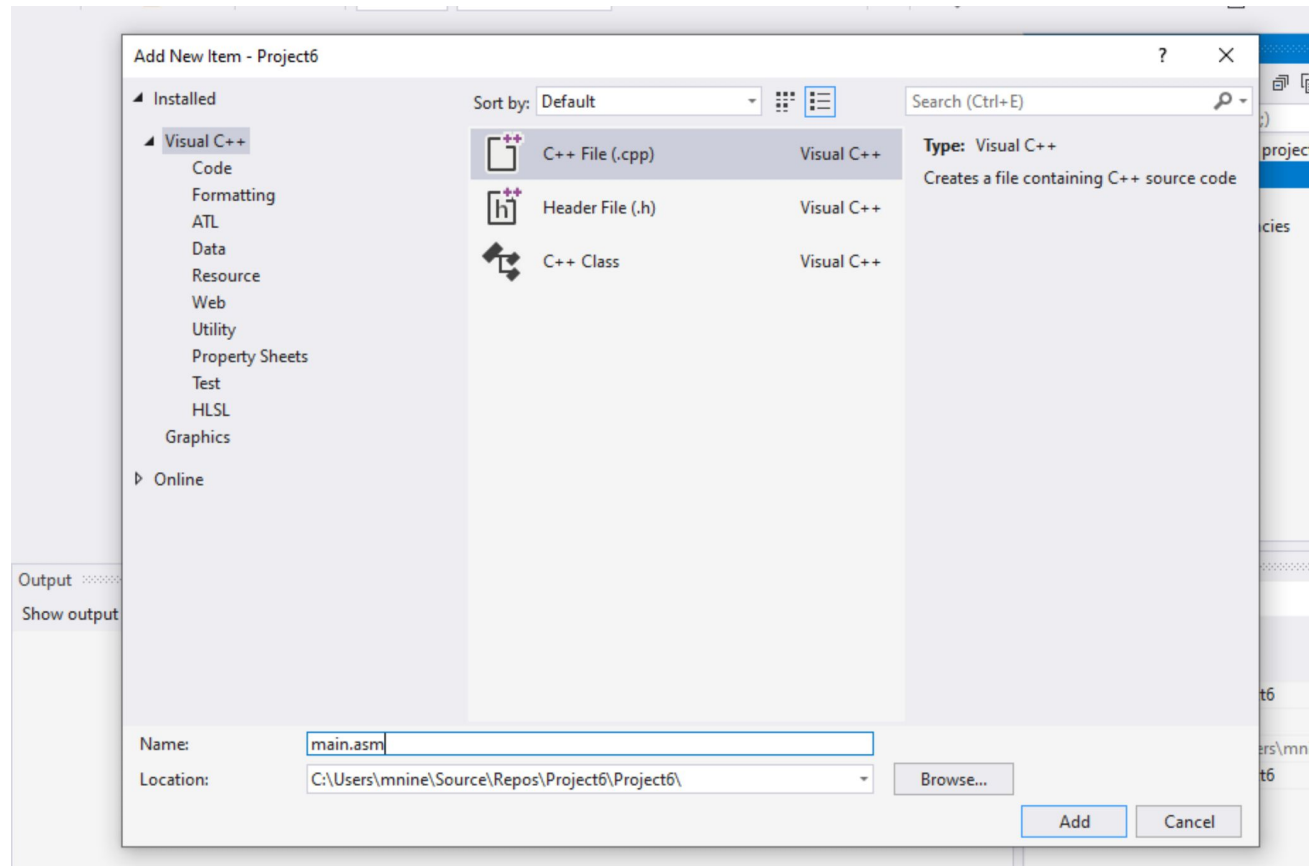


Step 1: Create a project (11)

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.

