CSC 3210

Computer Organization and Programming

Lab Work 6

Dr. Zulkar Nine

mnine@gsu.edu

Georgia State University

Fall 2021

Learning Objective

- Variable declaration
- Data types
- Comments
- String
- DUP operator

- Data Transfer, and 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!

Examples

Don’t need to turn in!

Example 1

- Which one of the following is a valid identifier ?
  - _work_var_1
  - #work_var_1
  - _1work_var
  - ?work_var_1

Example 1 Answer

- Which one of the following is a valid identifier ?
  - _work_var_1 (correct)
  - #work_var_1
  - _1work_var (correct)
  - ?work_var_1 (correct)

Example 2

```
.data
VAR word 10
.code
mov var, 15
```

This code will issue an error, because, identifiers are case sensitive. (True/False)

Example 2 answer

- No errors
- Identifiers are NOT case sensitive

Example 3

- The following is a valid multi-line comment :

```
COMMENT !
        This is a comment!
         This is a comment!
!
```

Example 3 Answer

- Character '!' is invalid, because it is present in the comment.
- Choose a character that is not present in the comment

Example 4

- Which of the following variable declarations are valid?
  - Val DW 12
  - Val SDWORD -12
  - Val WORD ABCDh

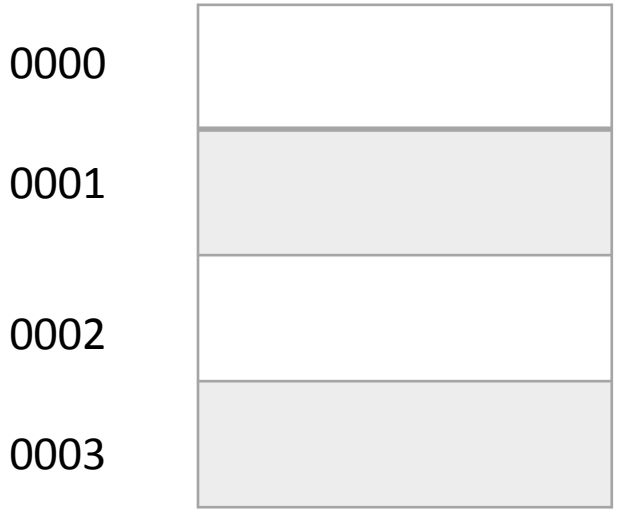
Example 4 answer

- Val DW 12 (valid)
- Val SDWORD -12. (valid)

- Val WORD ABCDh. (NOT Valid)
  - Correct declaration : var WORD 0ABCDh

Example 5

- What is the little endian representation of the following variable
  - Var DWORD 87654321h
  - Given that the first address assigned to the variable is 0000.



Example 5 Answer

- What is the little endian representation of the following variable
  - Var DWORD 87654321h

0000	**21**
0001	43
0002	65
0002	87

Lab Work 6 Instructions

- Lab 6(a) : Write a program to evaluate an expression with variables (4 Points)
- Lab 6(b): Write a program to see the data items in an array (3 points)
- Lab 6(c): Write a program to find the size of the string (3 points)

Due Date: Posted on iCollege

Submission Instruction

- There is an answer sheet provided with the lab
- Fill out the answer sheet and submit it to iCollege.

Lab 6(a)

Submission

Data Transfer and Arithmetic Instructions

- Write a program to implement the following expression in assembly:
  - **EAX = Xval - (Yval + Zval)**
    - Xval is signed 32-bit integer variable
    - Yval and Zval are unsigned 32-bit integer variable
    - Assign Xval 26, Yval 30, and Zval 40.
- How to approach this problem and its solution are provided in the upcoming slides.
- Make sure you understand the solution

Data Transfer and Arithmetic (add and sub) Instructions

- Before writing a program you should know what an instruction can and can not do
- **Add** and **subtract** can only do the following:
  - Add or sub register to/from register:

    add eax, ebx
  - Add or sub register to/from memory:

    sub mem1, eax
  - Add or sub memory to/from register:

    sub eax, mem1
  - Add or subtract immediate to/from memory:

    add mem1,3
  - Add or subtract immediate to/from register:

    add eax,3

Data Transfer and Arithmetic (add and sub) Instructions

- Before writing a program you should know what an instruction can and can not do
- **Add** and **subtract** can NOT do the following
  - Add or sub memory to/from memory:

    sub mem1,mem2 or add mem1,mem2

- The same apply for **mov** instruction

Data Transfer and Arithmetic (add and sub) Instructions

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

```
.386
.model flat, stdcall
.stack 4096
ExitProcess PROTO, dwExitCode: DWORD
; EAX = Xval - (Yval + Zval)
; parenthesis have higher precedence, do them first
.data
        Xval SDWORD 26
        Yval DWORD 30
        Zval DWORD 40
.code
main proc
; second term: (Yval + Zval)
        mov ebx,Yval
        add ebx,Zval
; sub the terms and store:
        sub Xval,ebx
        mov eax,Xval
        invoke ExitProcess, 0
main ENDP
END main
```

Submission Instruction

- There is an answer sheet attached to the lab
- 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 6(b)

Submission

Defining Data: **Using the DUP Operator**

- Use **DUP** to allocate (create space for) an array or string.
- Syntax: *counter* **DUP (*argument*)**
- *Counter* and *argument* must be constants or constant expressions

```
var1 BYTE 20 DUP(0) ; 20 bytes, all equal to zero
var2 BYTE 20 DUP(?) ; 20 bytes, uninitialized
var3 BYTE 4 DUP("STACK")   ; 20 bytes: "STACKSTACKSTACKSTACK"
var4 BYTE 10,3 DUP(0),20   ; 5 bytes
```

Dup Operator

- **Create a new application to run the following program.**
- **Build and run the program using the debugger**
- **Examine the content of the register AX**

```
; summation of the list values.
.386
.model flat, stdcall
.stack 4096
ExitProcess proto, dwExitCode:dword

.data

myWord word 4 dup(1,2,3,4,5)

.code

main proc

mov eax,0                    ; zeroing eax
```

```
mov ax,myWord+0        ;sum up the list values

add ax, myWord + 2
add ax, myWord + 4
add ax, myWord + 6
add ax, myWord + 8

invoke ExitProcess,0
main endp
end main
```

Note: Section Appendix has a reference to how to create a new project.

Submission Instruction

- Debug the code
- Answer the questions in the answer sheet.
  - What is the total size of the myWord array?
  - Debug the code until the 'invoke ExitProcess, 0'. Attach screenshot showing the content of AX register at the end.

Lab 6(c)

Submission

$ Operator

- **Create a new application to run the following program.**
- **Build and run the program using the debugger**
- **Examine the content of the register AL**

  (convert the value to decimal to see length of the string)

```
; Calculating the size of the String
.386
.model flat, stdcall
.stack 4096
ExitProcess proto, dwExitCode:dword
.data
myString byte "This is a very long string made by your instructor to test how $ works in this lab hope you will like it"
myString_length = ($ - myString)
.code
main proc
        mov eax,0
        mov al, myString_length

invoke ExitProcess,0
main endp
end main
```

Note: Section Appendix has a reference to how to create a new project.

Submission Instruction

- Debug the code
- Answer the questions in the answer sheet.
  - What is the difference between symbolic constant and variables?
  - Debug the code until 'invoke ExitProcess, 0'. Attach the screenshot showing the content of al register at the end.

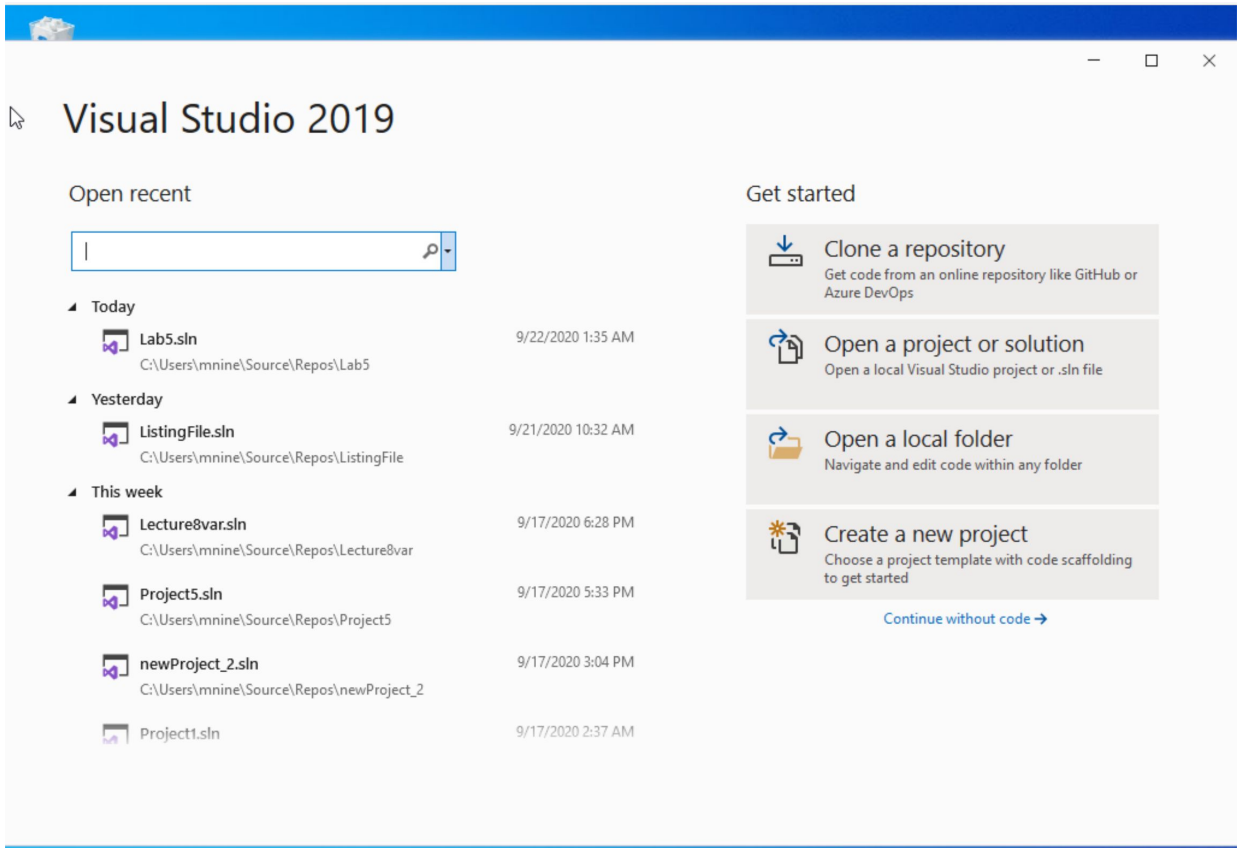
Appendix

Create a Project

Step 1: Create a project (1)

(1) Start Visual Studio

(2) Click Create a new Project

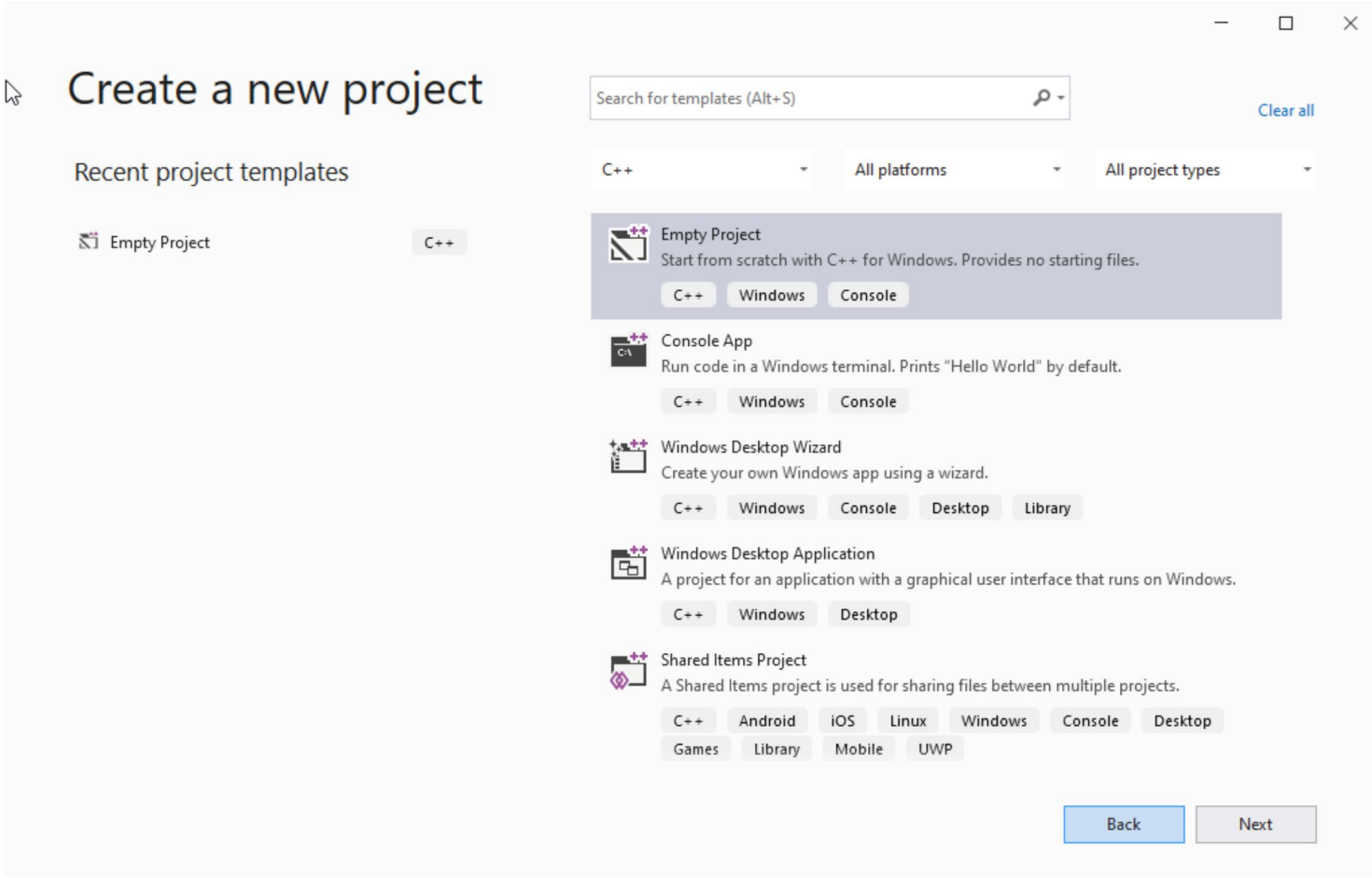


Step 1: Create a project (2)

(1) Select C++ as language

(2) Select Empty Project

(3) Click Next

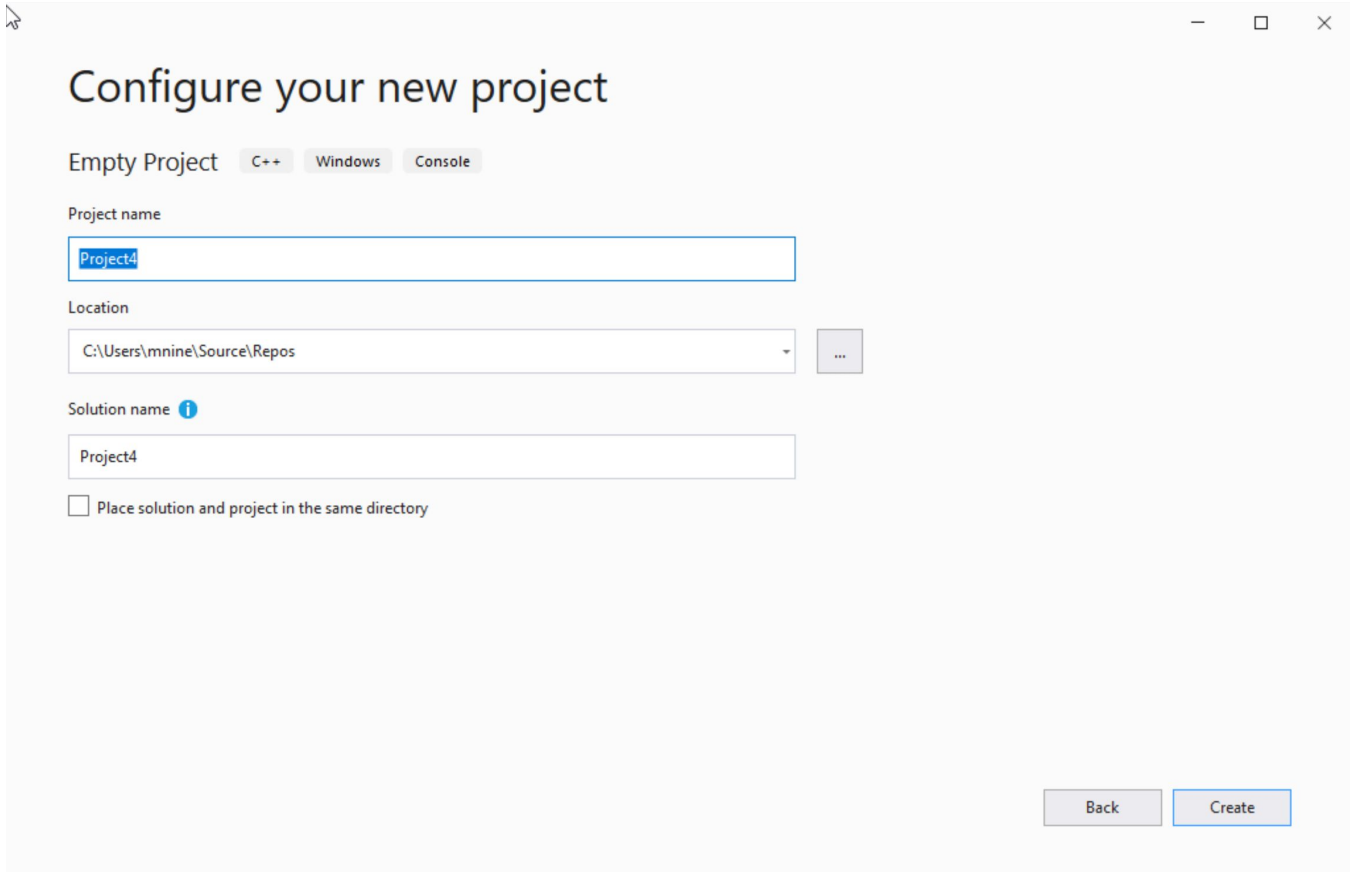


Step 1: Create a project (3)

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

(1) Also, you can change the project location

(2) Click Next

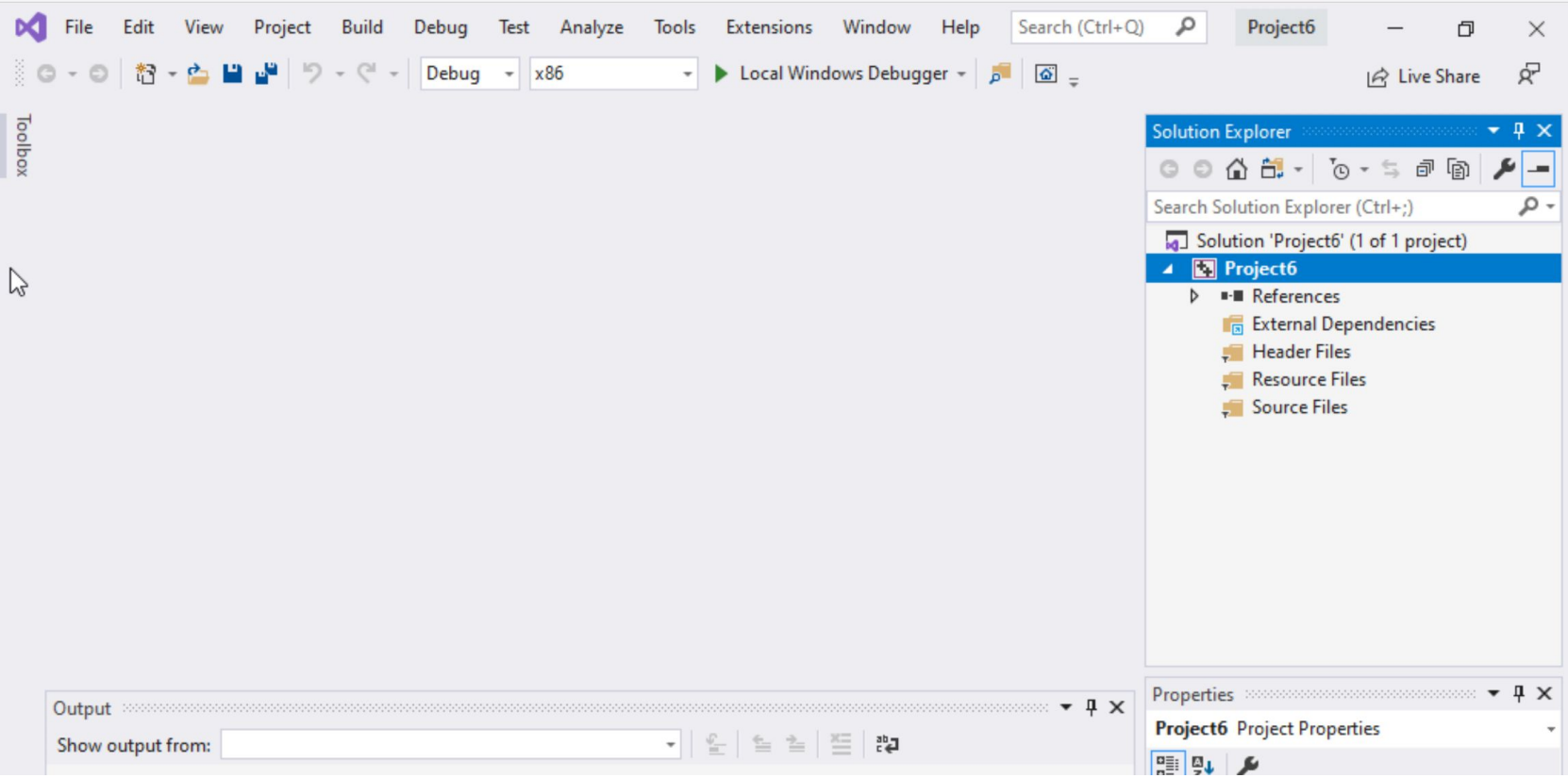


Step 1: Create a project (4)

Delete the

Following folders:

- Header files
- Resources Files, and
- Source Files



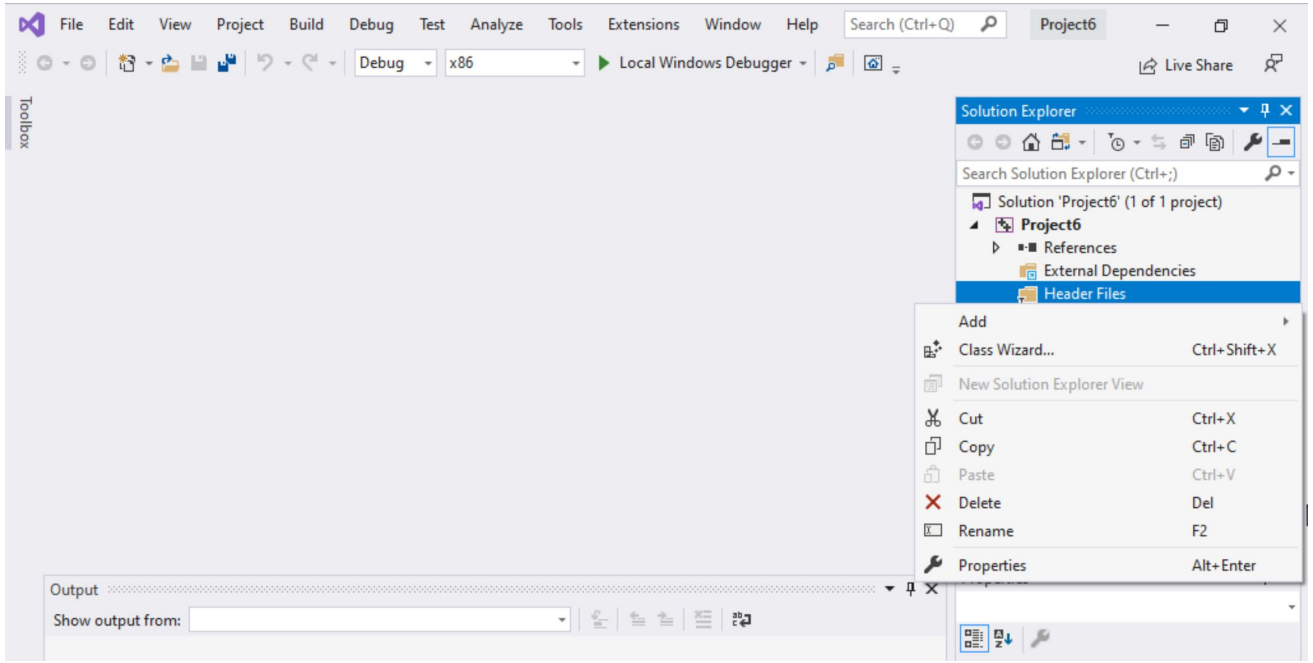
Step 1: Create a project (5)

To delete :

Select the folders

Right click on it

Select delete



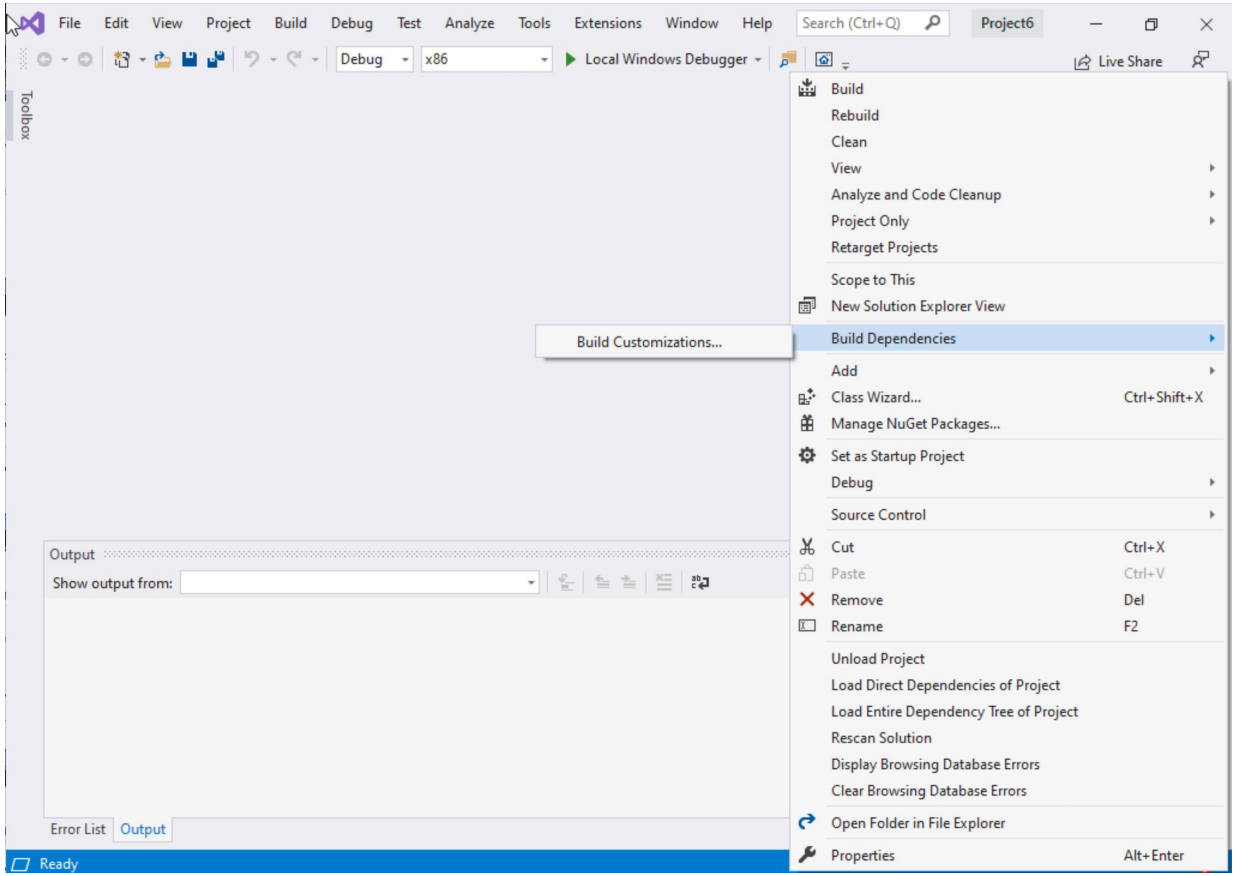
Step 1: Create a project (6)

Select Project Name on solution explorer

Right click on it

Go to Build Dependencies

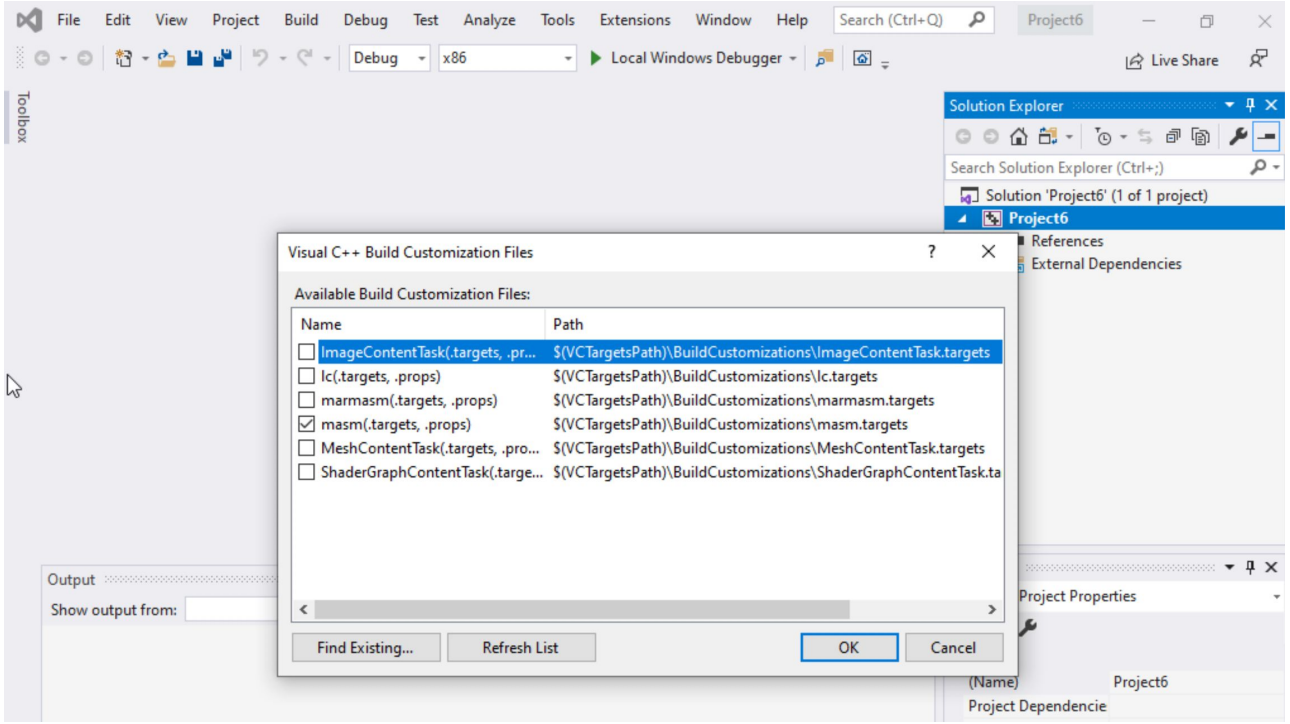
Click on Build Customizations



Step 1: Create a project (7)

Select masm(.target, .props)

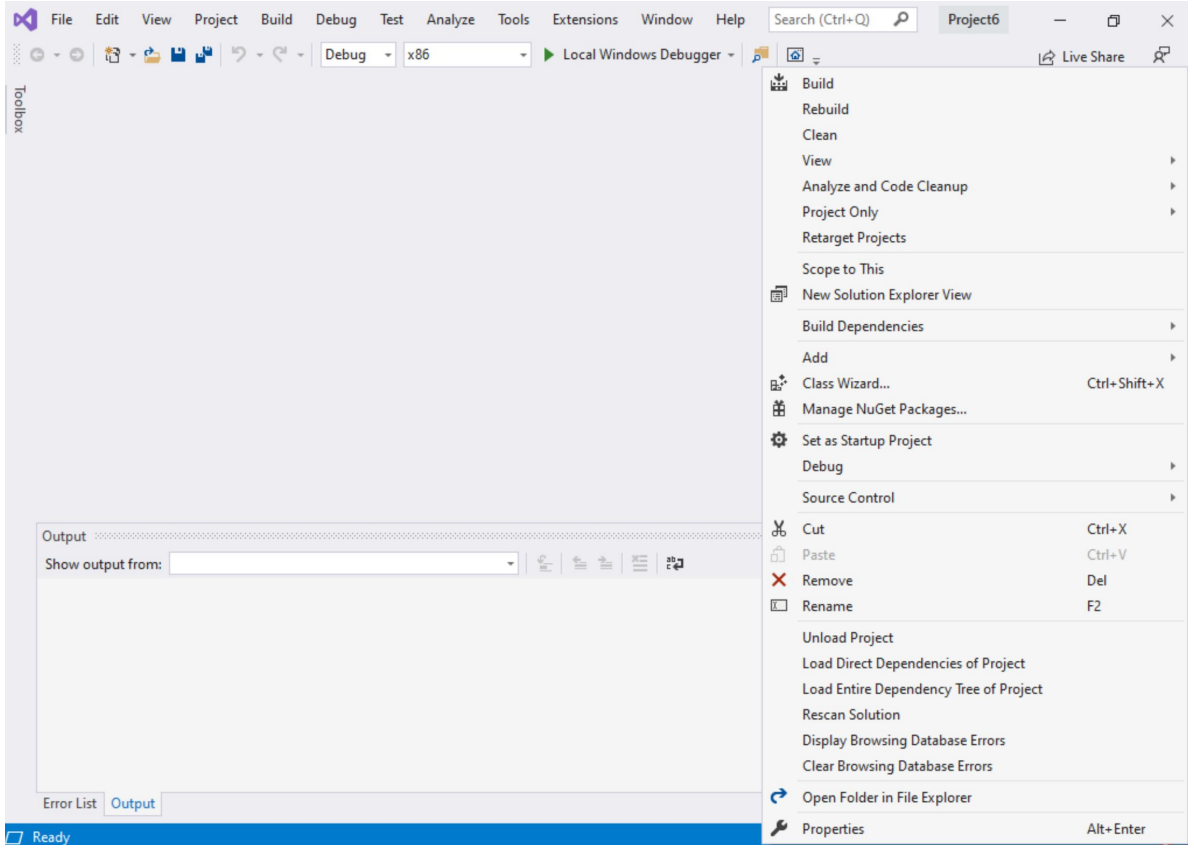
Click ok



Step 1: Create a project (8)

Right click on the Project name in the solution explorer

Click properties



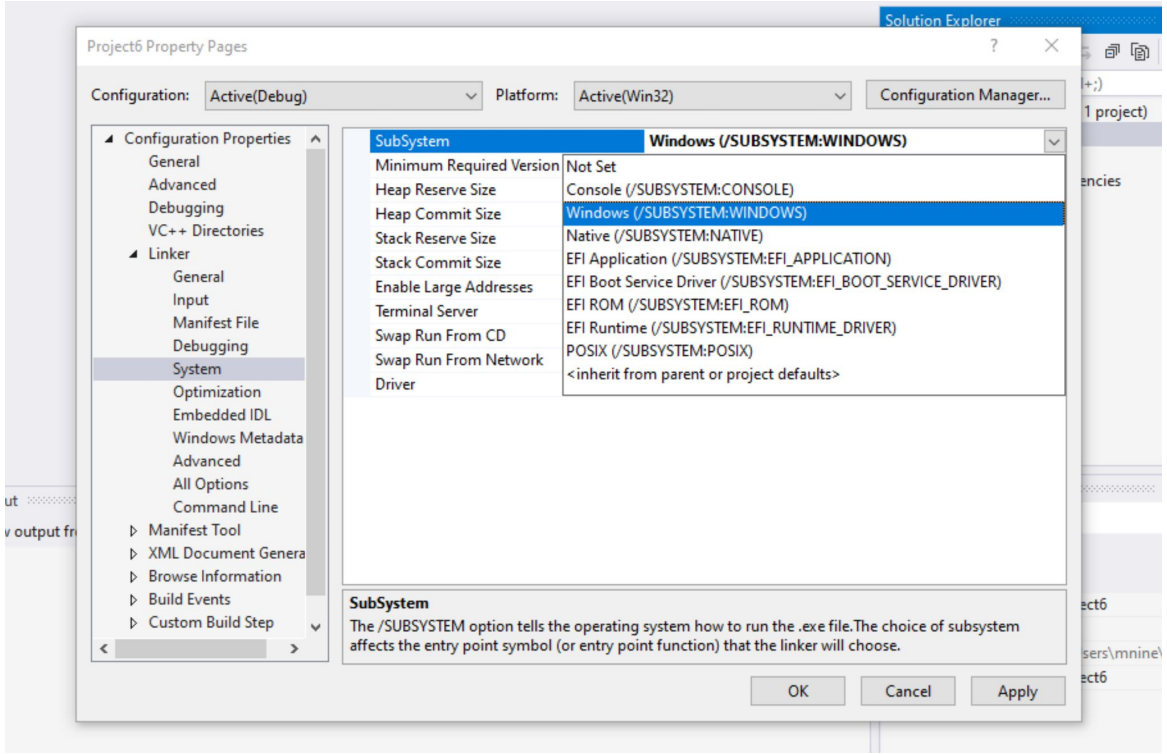
Step 1: Create a project (9)

Expand the ‘Linker’

Select ’System’

Select Windows(/SUBSYSTEM:WINDOWS)

Click OK



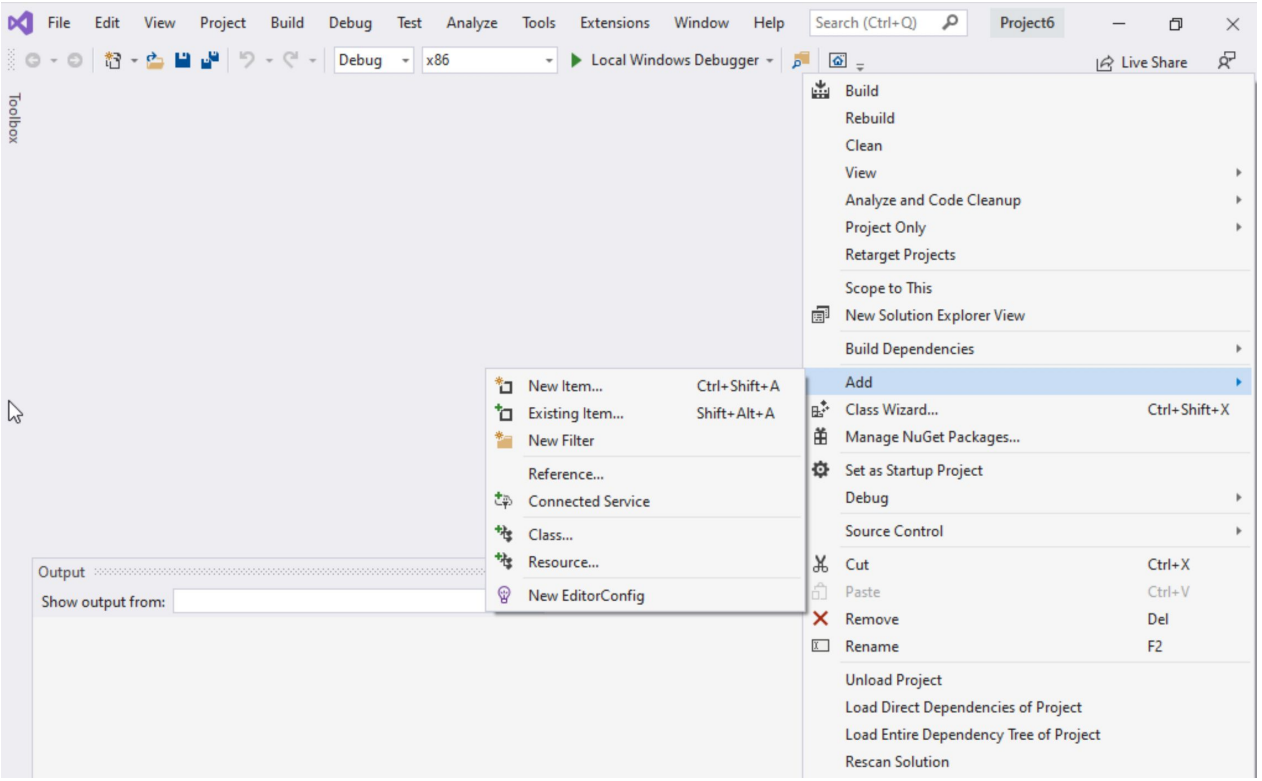
Step 1: Create a project (10)

Select Project name on solution explorer

Right click on it

Expand Add

Choose New Item

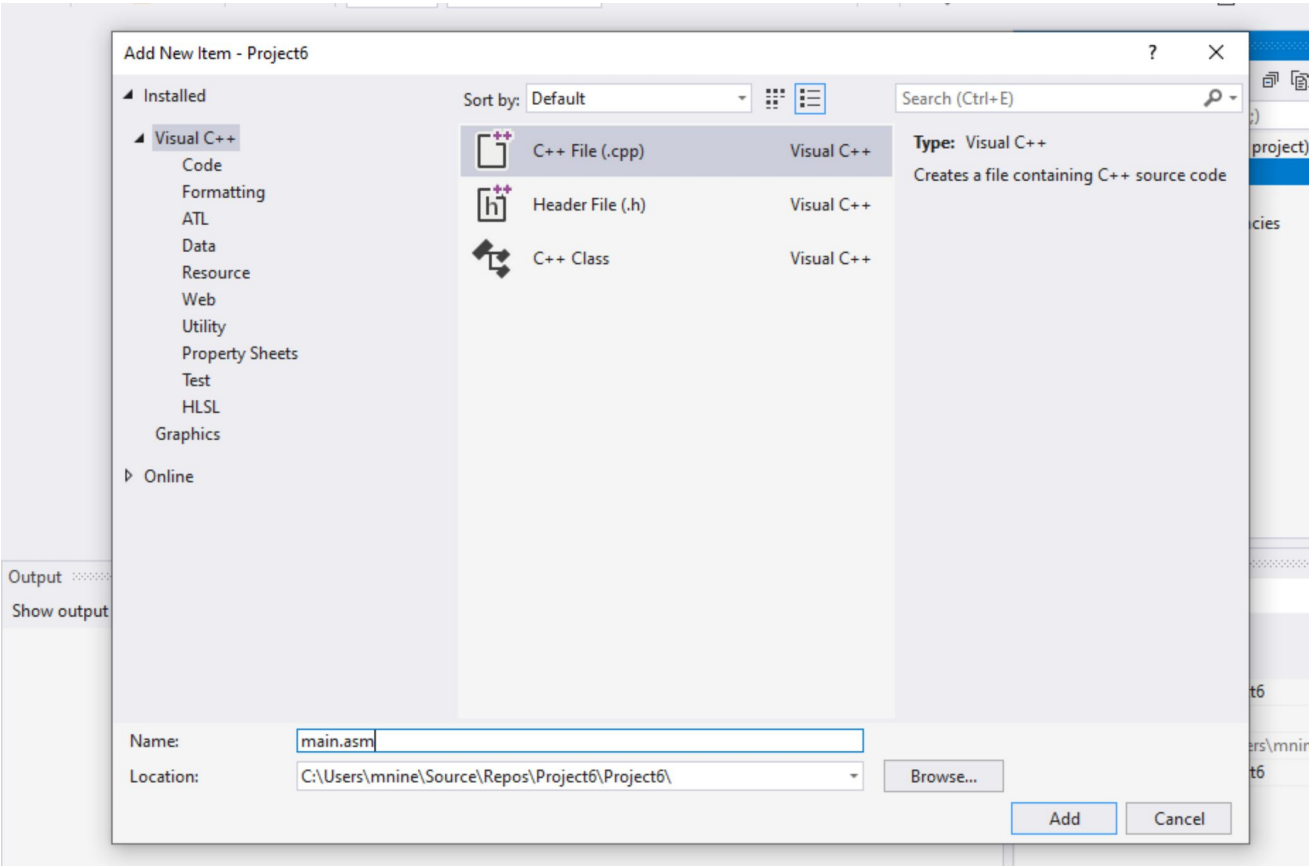


Step 1: Create a project (11)

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.